

**LAPORAN AKHIR**  
**HIBAH PENELITIAN UNGGULAN**



**PEMBANGUNAN *SISTEM INFORMASI GEORAFIS* SEBARAN  
FLORA, FAUNA, DAN AKTIVITAS KONSERVASI DALAM  
MENDUKUNG KELESTARIAN ALAM BERBASIS  
MANAJEMEN *RESORT* PROVINSI BENGKULU**

**Oleh:**

**Asahar Johar T, M.Kom  
Edy Hermansyah, Ph.D  
Arie Vatesia, M.T.I  
Desi Andreswari, M.Cs**

**FAKULTAS TEKNIK  
UNIVERSITAS BENGKULU  
2013**

## DAFTAR ISI

HALAMAN JUDUL .....	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iv
MOTTO DAN PERSEMBAHAN .....	v
KATA PENGANTAR .....	vi
ABSTRAK .....	ix
ABSTRACT .....	x
DAFTAR ISI .....	xi
DAFTAR GAMBAR .....	xiv
DAFTAR TABEL .....	xvii
DAFTAR LAMPIRAN.....	xviii

### **BAB I PENDAHULUAN ..... 1**

1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian .....	3
1.5 Manfaat Penelitian .....	3
1.6 Sistematika Penulisan .....	4

### **BAB II LANDASAN TEORI ..... 5**

2.1 Manajemen Berbasis Resort.....	5
2.2 Kawasan Konservasi Bengkulu Beserta Ekosistemnya .....	8
2.3 Sistem Informasi Geografis (SIG).....	11
2.3.1 Konsep Dasar Sistem .....	11
2.3.2 Konsep Dasar Informasi.....	11
2.3.3 Konsep Dasar Sistem Informasi.....	12
2.3.4 Sejarah Pengembangan Sistem Informasi Geografis .....	14
2.3.5 Sistem Informasi Geografis.....	16
2.3.6 Subsystem Sistem Informasi Geografis .....	20
2.3.7 Konsep Model Data Spasial .....	21
2.3.8 Data Vektor .....	22
2.3.9 Data Raster .....	22
2.3.10 Kebutuhan Terhadap Sistem Informasi Geografis .....	22
2.3.11 Sistem Komputer Untuk SIG .....	23
2.3.12 SIG Sebagai Ilmu Multi Disiplin .....	24
2.3.13 Cakupan Aplikasi SIG.....	25
2.3.14 SIG Sebagai Infrastruktur Informasi .....	26
2.3.15 SIG Sebagai Pengambilan Keputusan.....	28
2.3.16 Arsitektur SIG Berbasis Web.....	29
2.4 Siklus Pengembangan Perangkat Lunak SIG Berbasis Web .....	30
2.5 Web Beserta Perangkat Pendukungnya .....	33
2.5.1 Sejarah Lahirnya Web.....	33
2.5.2 HTML.....	34
2.5.3 CSS ( <i>Cascading Style Sheet</i> ).....	33
2.5.4 PHP ( <i>PHP Hypertext Processor</i> ) .....	35

2.5.5 Web Service .....	36
2.5.6 MySQL.....	39
2.5.7 Google Maps API.....	39
2.6 Android.....	40
2.7 <i>Unified Modeling Language (UML)</i> .....	43
2.7.1 Tipe Diagram UML.....	43
2.8 Bagan Alir ( <i>Flowchart</i> ) .....	51
2.9 Hasil Penelitian Terkait.....	54
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>56</b>
3.1 Jenis Penelitian .....	56
3.2 Sumber Data .....	56
3.3 Metode Penelitian.....	57
3.3.1 Metode Pengumpulan Data .....	57
3.3.2 Model Pengembangan Sistem .....	57
3.4 Model Pengujian Sistem ( <i>Black Box</i> ).....	61
3.5 Bagan Alir Sistem ( <i>Flowchart</i> ) .....	62
3.6 Sarana Pendukung .....	67
3.7 Jadwal Penelitian.....	68
<b>BAB IV ANALISIS DAN PERANCANGAN SISTEM .....</b>	<b>70</b>
4.1 Definisi dan Analisis Kebutuhan .....	70
4.1.1 Identifikasi Masalah .....	70
4.1.2 Analisis Sistem yang ada .....	57
4.1.3 Analisis Kebutuhan Pengguna .....	72
4.1.4 Analisis Kebutuhan Sistem .....	73
4.2 Desain Sistem dan Perangkat Lunak .....	65
4.2.1 Perancangan Model UML .....	66
4.2.1.1 <i>Behaviour Diagrams</i> .....	74
4.2.1.2 <i>Structure Diagrams</i> .....	89
4.2.2 Perancangan <i>Database</i> .....	98
4.2.3 Perancangan Tabel <i>Relationship</i> .....	95
4.2.3 Perancangan <i>User Interface</i> .....	99
<b>BAB V HASIL DAN PEMBAHASAN .....</b>	<b>106</b>
5.1 Implementasi Sistem .....	106
5.1.1 Penulisan Kode Program (Coding) .....	106
5.2 Pengujian Sistem .....	141
5.2.1 Pengujian <i>white box</i> .....	141
5.2.2 Pengujian <i>black box</i> .....	165
5.3 Uji Kelayakan Sistem.....	176
<b>BAB VI KESIMPULAN DAN SARAN .....</b>	<b>181</b>
6.1 Kesimpulan .....	181
6.2 Saran .....	182
<b>DAFTAR PUSTAKA .....</b>	<b>183</b>
<b>LAMPIRAN .....</b>	<b>185</b>

## DAFTAR GAMBAR

Gambar 2.1 Peta Kawasan Konservasi Provinsi Bengkulu .....	9
Gambar 2.2 Peta Kawasan Hutan Provinsi Bengkulu.....	9
Gambar 2.3 Subsistem SIG .....	21
Gambar 2.4 Contoh Gambar Geospasial.....	21
Gambar 2.5 Infrastruktur Informasi GIS .....	27
Gambar 2.6 SIG yang ideal .....	26
Gambar 2.7 GIS untuk Pengambilan Keputusan .....	28
Gambar 2.8 Arsitektur SIG berbasis Web .....	29
Gambar 2.9 Siklus pengembangan SIG berbasis web .....	30
Gambar 2.10 Layer Web Service .....	38
Gambar 2.11 Lapisan arsitektur android secara umum.....	42
Gambar 2.12 Diagram-diagram pemodelan sistem pada UML .....	44
Gambar 2.13 <i>Use Case Diagram</i> .....	45
Gambar 2.14 <i>Class Diagram</i> .....	45
Gambar 2.15 <i>Statechart Diagram</i> .....	46
Gambar 2.16 <i>Sequence Diagram</i> .....	47
Gambar 2.17 <i>Collaboration Diagram</i> .....	48
Gambar 2.18 <i>Activity Diagram</i> .....	49
Gambar 2.19 <i>Component Diagram</i> .....	50
Gambar 2.20 <i>Deployment Diagram</i> .....	51
Gambar 2.21 Notasi Bagan Alir Sistem .....	52
Gambar 2.22 Notasi Bagan Alir Program .....	53
Gambar 2.23 Notasi Bagan Alir Proses .....	54
Gambar 3.1 Model <i>Waterfall</i> .....	60
Gambar 3.2 <i>Flowchart</i> sistem operator resort dan petugas lapangan .....	64
Gambar 3.3 <i>Flowchart</i> sistem admin dan petugas lapangan.....	65
Gambar 3.4 <i>Flowchart</i> sistem admin dan operator resort.....	66
Gambar 3.5 <i>Flowchart</i> sistem admin dan masyarakat .....	67
Gambar 4.1 Peta Kawasan Hutan meliputi kawasan Konservasi .....	72
Gambar 4.2 <i>Use Case Diagram</i> .....	76
Gambar 4.3 <i>Activity Diagram Web</i> .....	77
Gambar 4.4 <i>Activity Diagram Aplikasi Mobile</i> .....	78
Gambar 4.5 <i>Sequence Diagram Web</i> .....	79
Gambar 4.6 <i>Sequence Diagram</i> Manajemen Konservasi dan Petugas .....	80
Gambar 4.7 <i>Sequence Diagram</i> Aktivitas Konservasi .....	81
Gambar 4.8 <i>Communication Diagram Aplikasi</i> .....	82
Gambar 4.9 <i>Communication Diagram Operator Resort</i> .....	82
Gambar 4.10 <i>Communication Diagram Admin</i> .....	83
Gambar 4.11 <i>Interaction Overview Diagram Admin</i> .....	83
Gambar 4.12 <i>Interaction Overview Diagram Operator</i> .....	84
Gambar 4.13 <i>Interaction Overview Diagram Petugas</i> .....	85
Gambar 4.14 <i>Timing Diagram</i> .....	86
Gambar 4.15 <i>Behavior State Machine Diagram</i> .....	87
Gambar 4.16 <i>Sub State</i> Tambah Data .....	88
Gambar 4.17 <i>Sub State</i> Edit Data.....	88

Gambar 4.18 <i>Sub State Sinkronisasi Data</i> .....	89
Gambar 4.19 <i>Protocol State Machine Diagram</i> .....	90
Gambar 4.20 <i>Class Diagram</i> .....	91
Gambar 4.21 <i>Object Diagram</i> .....	92
Gambar 4.22 <i>Package Diagram</i> .....	93
Gambar 4.23 <i>Component Diagram Aplikasi Mobile</i> .....	94
Gambar 4.24 <i>Component Diagram Web</i> .....	94
Gambar 4.25 <i>Deployment Diagram</i> .....	95
Gambar 4.26 <i>Perancangan Database</i> .....	100
Gambar 4.27 <i>Form Awal Aplikasi Mobile Konservasi</i> .....	101
Gambar 4.28 <i>Form Awal Aplikasi Mobile Konservasi</i> .....	102
Gambar 4.29 <i>Form Aktivitas Konservasi Sebaran Flora dan Fauna</i> .....	103
Gambar 4.30 <i>Form Aktivitas Konservasi</i> .....	103
Gambar 4.31 <i>Form Peta</i> .....	104
Gambar 4.32 <i>Form Utama Web</i> .....	104
Gambar 4.33 <i>Form Utama Admin</i> .....	105
Gambar 4.34 <i>Form Utama Operator</i> .....	106
Gambar 5.1 <i>IDE Eclipse Indigo</i> .....	109
Gambar 5.2 <i>Struktur XML Tampilan Password</i> .....	110
Gambar 5.3 <i>Struktur XML Tampilan Beranda Menu</i> .....	111
Gambar 5.4 <i>Struktur XML Sinkronisasi</i> .....	119
Gambar 5.5 <i>Instalasi Google Play Services</i> .....	120
Gambar 5.6 <i>Import Library Google Play Services</i> .....	120
Gambar 5.7 <i>Import Library Google Play Services ke Project</i> .....	121
Gambar 5.8 <i>Menu Service Google Maps API</i> .....	121
Gambar 5.9 <i>API Key Android dan Browser</i> .....	122
Gambar 5.10 <i>Android Virtual Devices (AVD)</i> .....	143
Gambar 5.11 <i>Login (AVD)</i> .....	144
Gambar 5.12 <i>Menu (AVD)</i> .....	145
Gambar 5.13 <i>Menu Aktivitas Konservasi (AVD)</i> .....	145
Gambar 5.14 <i>Submenu Temuan (AVD)</i> .....	146
Gambar 5.15 <i>Submenu Pelanggaran (AVD)</i> .....	147
Gambar 5.16 <i>Submenu Kejadian (AVD)</i> .....	148
Gambar 5.17 <i>Submenu Tanda Batas (AVD)</i> .....	149
Gambar 5.18 <i>Submenu Penyuluhan (AVD)</i> .....	149
Gambar 5.19 <i>Submenu Sosialisasi (AVD)</i> .....	150
Gambar 5.20 <i>Submenu Koordinasi (AVD)</i> .....	150
Gambar 5.21 <i>Submenu Pemasangan Papan Peringatan (AVD)</i> .....	151
Gambar 5.22 <i>Sinkronisasi (AVD)</i> .....	152
Gambar 5.22 <i>Sinkronisasi (Device Android)</i> .....	153
Gambar 5.22 <i>Sinkronisasi (Device Android)</i> .....	154
Gambar 5.23 <i>Beranda (Browser Mozilla Firefox)</i> .....	155
Gambar 5.24 <i>Aktivitas (Browser Mozilla Firefox)</i> .....	155
Gambar 5.25 <i>Petugas (Browser Mozilla Firefox)</i> .....	156
Gambar 5.26 <i>Resort (Browser Mozilla Firefox)</i> .....	156
Gambar 5.27 <i>TallySheet (Browser Mozilla Firefox)</i> .....	157
Gambar 5.28 <i>Peta (Browser Mozilla Firefox)</i> .....	158
Gambar 5.29 <i>Rute (Browser Mozilla Firefox)</i> .....	159

Gambar 5.29 Rute ( <i>Browser Mozilla Firefox</i> ) .....	160
Gambar 5.31 Pengaduan ( <i>Browser Mozilla Firefox</i> ) .....	161
Gambar 5.32 Antarmuka <i>Operator</i> ( <i>Browser Mozilla Firefox</i> ).....	162
Gambar 5.33 Antarmuka <i>User</i> ( <i>Browser Mozilla Firefox</i> ).....	163
Gambar 5.34 Menu <i>Resort</i> ( <i>Browser Mozilla Firefox</i> ).....	164
Gambar 5.35 <i>Menu Aktivitas Konservasi</i> ( <i>Browser Mozilla Firefox</i> ).....	164
Gambar 5.36 Menu <i>Galeri</i> ( <i>Browser Mozilla Firefox</i> ) .....	165
Gambar 5.37 <i>Menu Autentifikasi</i> ( <i>Browser Mozilla Firefox</i> ) .....	165
Gambar 5.38 SubMenu <i>Lapak Pengaduan</i> ( <i>Browser Mozilla Firefox</i> ).....	166
Gambar 5.39 SubMenu <i>Seputar Konservasi</i> ( <i>Browser Mozilla Firefox</i> ).....	166
Gambar 5.40 Persamaan <i>skala likert</i> .....	179
Gambar 5.41 Grafik <i>Presentase Hasil Angket Variabel 1</i> .....	180
Gambar 5.42 Grafik <i>Presentase Hasil Angket Variabel 2</i> .....	181
Gambar 5.43 Grafik <i>Presentase Hasil Angket Variabel 3</i> .....	182

## DAFTAR TABEL

Tabel 2.1 Kelebihan-kelebihan SIG .....	23
Tabel 3.1 Rencana Penelitian .....	70
Tabel 4.1 Struktur tabel tbl_bencana .....	96
Tabel 4.2 Struktur tabel tbl_provinsi .....	96
Tabel 4.3 Struktur tabel tb_aktivitas .....	96
Tabel 4.4 Struktur tabel tb_batas .....	96
Tabel 4.5 Struktur tabel tb_kawasan .....	97
Tabel 4.6 Struktur tabel tb_kejadian .....	97
Tabel 4.7 Struktur tabel tb_koordinasi .....	97
Tabel 4.8 Struktur tabel tb_pasangpapan .....	97
Tabel 4.9 Struktur tabel tb_pelanggaran .....	98
Tabel 4.10 Struktur tabel tb_penyuluhan .....	98
Tabel 4.11 Struktur tabel tb_resort .....	98
Tabel 4.12 Struktur tabel tb_sosialisasi .....	98
Tabel 4.13 Struktur tabel tb_temuan .....	98
Tabel 4.14 Struktur tabel tb_tem_mas .....	99
Tabel 4.15 Struktur tabel tb_user .....	99
Tabel 5.1 Pengujian Masuk Aplikasi .....	167
Tabel 5.2 Pengujian Beranda Menu .....	168
Tabel 5.3 Pengujian Menu Utama (uji normal) .....	168
Tabel 5.4 Pengujian Menu Aktivitas Konservasi .....	169
Tabel 5.5 Pengujian Menu Patroli .....	169
Tabel 5.6 Pengujian Menu Penyuluhan .....	170
Tabel 5.7 Pengujian Submenu Temuan (uji normal) .....	170
Tabel 5.8 Pengujian Submenu Pelanggaran .....	171
Tabel 5.9 Pengujian Submenu Kejadian .....	172
Tabel 5.10 Pengujian Submenu Tanda Batas .....	172
Tabel 5.11 Pengujian Submenu Penyuluhan .....	173
Tabel 5.12 Pengujian Submenu Sosialisasi .....	174
Tabel 5.13 Pengujian Submenu Koordinasi .....	174
Tabel 5.14 Pengujian Submenu Pemasangan Papan Peringatan .....	175
Tabel 5.15 Pengujian Sinkronisasi .....	176
Tabel 5.16 Pengujian Peta .....	176
Tabel 5.17 Pengujian Web .....	177
Tabel 5.18 Kategori Penilaian .....	179
Tabel 5.19 Hasil Penilaian Variabel Tampilan .....	180
Tabel 5.20 Hasil Penilaian Variabel Kemudahan Pengguna .....	181
Tabel 5.21 Hasil Penilaian Variabel Kinerja Sistem .....	182

## DAFTAR LAMPIRAN

Lampiran A Surat Ijin dan Selesai Penelitian .....	A-1
Lampiran B Pengujian <i>Method</i> dan <i>Atribut</i> Sistem ( <i>WhiteBox</i> ) .....	B-1
Lampiran C Pengujian Fungsional Sistem ( <i>BlackBox</i> ).....	C-1
Lampiran D Pengujian Waktu.....	D-1
Lampiran E Angket Uji Kelayakan Sistem.....	E-1
Lampiran F Tabulasi Data Angket .....	F-1
Lampiran G Hasil Wawancara .....	G-1
Lampiran H Tabel Aktivitas Konservasi.....	H-1
Lampiran I Data Kegiatan Konservasi.....	I-1
Lampiran J Daftar Gambar Kegiatan Konservasi .....	J-1



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Provinsi Bengkulu semula merupakan suatu karesidenan dalam wilayah Provinsi Sumatera Selatan, sejak tanggal 18 November 1968 diresmikan menjadi Provinsi Daerah Tingkat I Bengkulu dan merupakan Provinsi ke-26 di Republik Indonesia. Termasuk pula dalam wilayah Provinsi Bengkulu yaitu Pulau Enggano, Pulau Tikus dan Pulau Mega. Secara geografis Provinsi Bengkulu terletak pada 101' 01" dan 103' 41" Bujur Timur serta 20' 16" dan 3' 31" Lintang Selatan dengan ketinggian dari permukaan laut 0-20 m (Bengkulu) sampai dengan 627-733 (Curup). Provinsi ini terletak di Pantai Barat Pulau Sumatera, membujur dari Utara ke Selatan, di antara Bukit Barisan di sebelah Timur dan Samudera Indonesia di sebelah Barat dengan luas wilayah lebih kurang 21.089,38 km<sup>2</sup> atau 2.108.938 ha dengan batas-batas sebagai berikut : sebelah utara berbatasan dengan Provinsi Sumatera Barat, sebelah timur berbatasan dengan Provinsi Jambi dan Provinsi Sumatera Selatan, sebelah selatan berbatasan dengan Samudera Indonesia dan Provinsi Lampung dan sebelah barat berbatasan dengan Samudera Indonesia.

Dari luas wilayah tersebut, 63,1% merupakan lahan budidaya dan 36,9% merupakan lahan non-budidaya yang tetap dilestarikan sebagai kawasan lindung untuk konservasi yang tidak boleh diganggu dan diambil manfaatnya. Hutan tropis Provinsi Bengkulu memiliki sumber kekayaan flora dan fauna yang dapat dikembangkan menjadi objek wisata. Kekayaan flora hutan tropis yang sudah terkenal dan telah menjadi objek wisata hutan adalah bunga *Rafflesia Arnoldi*, yang terdapat di hutan Kabupaten Bengkulu Utara, disamping bunga *Vanda Hookeriana*, kayu Meranti, dan lain-lain. Sementara kekayaan Fauna yang berpotensi untuk dikembangkan menjadi objek wisata adalah harimau sumatera, rusa, gajah, dan kerbau liar.

Penataan sampai ke *resort* akan berimplikasi perubahan tugas pokok Polisi Hutan, bukan sekedar melakukan pengamanan/patroli kawasan, tetapi akan muncul tugas-tugas tambahan, berinteraksi dengan masyarakat di desa-desa yang berbatasan dengan kawasan. Ilmu-ilmu sosial, antropologi budaya, dan resolusi konflik akan diperlukan dalam memberikan pembekalan kepada Polhut maupun staf fungsional (PEH) di kawasan konservasi. Perubahan revolusioner maupun bertahap akan mendorong organisasi UPT (Unit Pelaksana Teknis) dan Pusat untuk mereposisi perannya menjadi organisasi yang proaktif dan bukan sekedar organisasi yang reaktif sebagai sekedar pemadam kebakaran. Organisasi proaktif akan berusaha mencegah terjadinya sesuatu, bukan sekedar melaksanakan hukum secara parsial dan cenderung salah sasaran. Perlu adanya keseimbangan antara kutub ekosentrisme dan kutub antroposentrisme dalam pengelolaan kawasan konservasi. Ilmu ekologi manusia juga akan sangat bermanfaat dalam membantu membentuk pola-pola pengelolaan kawasan konservasi yang berwajah humanis, ramah, dan inklusif.

Dari data dan fakta yang diungkapkan pada paragraf sebelumnya dapat terlihat pentingnya inventarisasi data flora, fauna, dan aktivitas konservasi di provinsi Bengkulu untuk mendukung kelestarian alam dan lingkungan hidup. Sistem Informasi Geografis ini nantinya akan bermanfaat untuk lembaga-lembaga terkait yang ikut andil dan berperan dalam menjaga kelestarian alam dan lingkungan hidup di provinsi Bengkulu. Oleh karena itu, penelitian ini akan membangun **“Sistem Informasi Georafis Sebaran Flora, Fauna, dan Aktivitas Konservasi Dalam Mendukung Kelestarian Alam Berbasis Manajemen *Resort* Provinsi Bengkulu”**.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang masalah yang telah dikemukakan diatas, maka rumusan masalah yang muncul adalah sebagai berikut:

1. Bagaimana membangun Sistem Informasi Georafis Sebaran Flora, Fauna, Dan Aktivitas Konservasi Dalam Mendukung Kelestarian Alam Berbasis Manajemen *Resort* Provinsi Bengkulu?

2. Bagaimana implementasi, dukungan dan relevansi sistem ini terhadap kelestarian alam dan lingkungan hidup di provinsi Bengkulu?

### **1.3 Batasan Masalah**

Ruang lingkup permasalahan yang akan dibahas dalam penelitian ini sebagai berikut:

1. Berdasarkan keterangan data sekunder yang diperoleh dari Balai Konservasi Sumber Daya Alam dan Dinas Kehutanan Provinsi Bengkulu, wilayah yang akan dikelola meliputi *Resort* yang ada di Provinsi Bengkulu.
2. Penelitian merujuk kepada sebaran flora, fauna dan aktivitas konservasi di masing-masing *resort* yang ada di Provinsi Bengkulu .
3. Pembangunan Sistem Informasi Georafis Sebaran Flora, Fauna, dan Aktivitas Konservasi Dalam Mendukung Kelestarian Alam Berbasis Manajemen *Resort* Provinsi Bengkulu menggunakan Smartphone Android sebagai media pengumpulan data dan perancangan Website GIS sebagai media pengelolaan dengan menggunakan bahasa pemrograman PHP dan MySQL.

### **1.4 Tujuan Penelitian**

Tujuan dari penelitian ini adalah sebagai berikut :

1. Membangun Sistem Informasi Georafis Sebaran Flora, Fauna, Dan Aktivitas Konservasi Dalam Mendukung Kelestarian Alam Berbasis Manajemen *Resort* Provinsi Bengkulu.
2. Mengimplementasikan dukungan dan relevansi sistem ini terhadap kelestarian alam dan lingkungan hidup di Provinsi Bengkulu.

### **1.5 Manfaat Penelitian**

Manfaat utama dari penelitian ini adalah sebagai berikut:

1. Bengkulu memiliki Sistem Informasi Geografis yang memiliki basis data flora, fauna, dan aktivitas konservasi yang dapat digunakan oleh lembaga terkait untuk mendukung kelestarian alam.

2. Provinsi Bengkulu memiliki sistem basis data yang dapat membantu pengambil kebijakan untuk mengambil keputusan yang tepat dalam menjaga kelestarian alam Bengkulu dengan data yang akurat.
3. Menduniakan pengetahuan sumber daya Bengkulu yang menyangkut flora, fauna, dan aktivitas konservasi.

## **1.6 Sistematika Penulisan**

Laporan penelitian ini tersusun dari enam bab, sesuai panduan Tugas Akhir yang diberikan dari Program Studi Teknik Informatika Universitas Bengkulu, secara garis besar akan dijelaskan sebagai berikut :

### **BAB I PENDAHULUAN**

Bab I berisi tentang latar belakang yang memunculkan suatu permasalahan terkait, sehingga dibutuhkan sebuah penyelesaian untuk mengatasinya. Pada bab ini juga berisi latar belakang, rumusan masalah, tujuan, manfaat dan sistematika penulisan tugas akhir.

### **BAB II LANDASAN TEORI**

Pada bab ini akan dijelaskan mengenai tinjauan pustaka yang berisikan teori-teori secara garis besar yang berhubungan dengan penelitian.

### **BAB III METODE PENELITIAN**

Pada bab ini berisi tentang metode-metode yang digunakan dalam penelitian, seperti teknik pengumpulan data, metode pengembangan sistem, metode pengujian dan jadwal penelitian.

### **BAB IV ANALISIS DAN PERANCANGAN SISTEM**

Bab ini menjelaskan tentang tahapan analisis dan perancangan sistem yang akan dibangun meliputi perancangan sistem, analisis sistem, desain sistem dan implementasi sistem.

### **BAB V HASIL DAN PEMBAHASAN**

Bab ini menjelaskan serta menguraikan hasil perancangan dan implementasi sistem yang telah dibuat.

### **BAB VI PENUTUP**

Bab ini merupakan bab terakhir yang berisi kesimpulan dan saran dari penelitian yang telah dilakukan serta harapan terhadap pengembangan sistem kedepannya.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Manajemen Berbasis Resort**

Kawasan konservasi berdasarkan Undang-Undang Nomor 5 Tahun 1990 tentang Konservasi Keanekaragaman Hayati dan Ekosistemnya adalah kawasan yang ditetapkan oleh pemerintah sebagai kawasan suaka alam yaitu cagar alam dan suaka margasatwa, kawasan pelestarian alam yaitu taman nasional, taman wisata alam dan taman hutan raya, dan taman buru. Berdasarkan Permenhut P.13/Menhut-II/2005 tanggal 6 Mei 2005, Direktorat Jenderal PHKA mengembang tugas untuk melakukan pengelolaan kawasan konservasi seluas 27.190.993 hektar. Direktorat Jenderal PHKA memberikan tugas pengelolaan kepada unit-unit pelaksana teknis untuk melakukan pengelolaan kawasan konservasi guna menjamin kelestarian keanekaragaman hayati dan ekosistemnya di wilayah kerjanya masing-masing.

Secara absolut, Unit Pelaksana Teknis dalam melakukan pengelolaan kawasan konservasi akan menjaga dan melestarikan keanekaragaman hayati dan ekosistemnya. Secara ilmiah, menjaga dan melestarikan keanekaragamann hayati dan ekosistemnya dilakukan dengan menciptakan kondisi sedemikian rupa sehingga *key features* kawasan dapat berproses secara alami serta dapat dimonitor dinamikanya, termasuk dengan melakukan tindakan konservasi yang dilakukan. Dalam pengelolaan kawasan konservasi muncul berbagai permasalahan seperti dihapusnya eselon V setingkat kepala *resort*, belum lengkapnya sistem kerja di tingkat *resort* dan tidak tersedianya anggaran kegiatan di *resort* mendorong ketidakjelasan pelaksanaan kegiatan minimal yang harus dilaksanakan di tingkat *resort*/lapangan. Hal ini lebih lanjut mengakibatkan meningkatnya ketidakhadiran staf di lapangan. Ketika lapangan ditinggalkan, maka kawasan seolah-olah menjadi tidak bertuan dan cenderung mengarah ke dalam situasi yang disebut sebagai “*open acces*” .

Pada situasi seperti inilah maka intensitas berbagai bentuk gangguan terhadap kawasan semakin meningkat. Gangguan tersebut terwujud dalam

berbagai kegiatan ilegal antara lain perambahan, penyerobotan, konflik batas, *illegal logging*, perburuan satwa dan pendudukan kawasan oleh berbagai pihak yang tidak bertanggungjawab. Gangguan ini akan semakin meningkat sebagai akibat rendahnya kehadiran staf di lapangan. Hal inilah yang mendasari perlunya pengelolaan kawasan konservasi berbasis resort sehingga kehadiran staf di lapangan meningkat dan kegiatan-kegiatan minimal yang harus dilaksanakan di tingkat lapangan terlaksana dengan baik dan fungsi pencegahan terhadap meluasnya gangguan hutan tercapai.

Penetapan suatu kawasan konservasi merupakan salah satu strategi konservasi dengan tujuan untuk melindungi keanekaragaman jenis dan ekosistemnya dari kepunahan. Dalam IUCN (*International Union for Conservation of Nature*) tahun 1994, kawasan konservasi didefinisikan sebagai wilayah daratan dan atau laut yang secara khusus diperuntukkan bagi perlindungan dan pemeliharaan keanekaragaman hayati, sumberdaya alam serta sumberdaya budaya, dikelola melalui cara-cara legal atau cara-cara efektif lainnya. Dalam paradigma lama, kawasan konservasi dianggap sebagai sebuah kawasan yang cenderung eksklusif dengan penekanan pada perlindungan sistem penyangga kehidupan serta pengawetan jenis tumbuhan dan satwa, sehingga salah satu tujuan pemanfaatan secara lestari sumberdaya alam hayati dan ekosistemnya belum tersentuh. Pemikiran ini menyebabkan rencana pengelolaan tersebut tidak mengakomodasi berbagai kepentingan yang terkait dengan keberadaan kawasan tersebut. Akibatnya adalah masyarakat di sekitar kawasan merasa tidak dilibatkan dalam pengelolaan kawasan konservasi.

Kemudian muncul paradigma baru yang menyatakan bahwa pengelolaan kawasan konservasi adalah rencana dengan prinsip pengelolaan yang terintegrasi dan mengakomodir berbagai kepentingan yang terkait, khususnya menyangkut kesejahteraan masyarakat di sekitar kawasan (IUCN, 2003). Konsep pelestarian modern adalah pemeliharaan dan pemanfaatan sumberdaya bumi secara bijaksana. Penetapan dan pengelolaan kawasan yang dilindungi adalah salah satu cara terpenting untuk mendapatkan jaminan agar sumberdaya alam dapat dilestarikan, sehingga sumberdaya tersebut dapat lebih memenuhi kebutuhan umat manusia sekarang dan di masa akan datang. Kawasan yang dilindungi apabila dirancang

dan dikelola secara tepat dapat memberikan keuntungan yang lestari bagi masyarakat. Pelestarian memegang peranan penting dalam pembangunan sosial ekonomi di lingkungan pedesaan dan turut mengembangkan peningkatan kesejahteraan dan kualitas hidup masyarakat penghuni pedesaan tersebut (MacKinnon et al, 1993).

Pembangunan kawasan konservasi di Indonesia masih dihadapkan pada berbagai gangguan dan ancaman yang menyebabkan kerusakan dan kawasan konservasi belum dapat berfungsi secara optimal. Berbagai bentuk gangguan dan ancaman terhadap kawasan konservasi adalah: pencurian dan penembangan liar, perambahan, peredaran dan perdagangan flora dan fauna secara ilegal, perburuan liar, penangkapan melebihi quota, dan penyelundupan flora dan fauna langka dan dilindungi. Keberhasilan pengelolaan kawasan banyak bergantung pada kadar dukungan dan penghargaan yang diberikan kepada kawasan yang dilindungi oleh masyarakat sekitarnya. Di tempat dimana kawasan yang dilindungi dipandang sebagai penghalang, masyarakat setempat dapat mengagalkan pelestariannya, tetapi bila dianggap sebagai sesuatu yang positif manfaatnya, masyarakat sendiri yang akan bekerjasama dengan pengelola dalam melindungi kawasan tersebut dari pengembangan yang membahayakan (MacKinnon et al, 1993)

Di dalam IUCN (2000) dikatakan bahwa untuk mendukung konsep pembangunan berkelanjutan dalam pengelolaan kawasan konservasi, IUCN dan WWF telah merekomendasikan:

1. Kawasan konservasi akan bertahan hanya dengan melihat nilai, dalam pengertian yang lebih luas, untuk kepentingan bangsa secara keseluruhan dan kepada masyarakat lokal pada khususnya.
2. Keberadaan masyarakat adat dan lokal dalam kawasan konservasi harus didukung penuh dengan bentuk partisipasi dalam bentuk *co-management* pengelolaan sumberdaya dan cara ini tidak akan mengikis sasaran hasil pengelolaan kawasan konservasi.
3. Pengetahuan, inovasi dan praktek yang diterapkan oleh masyarakat lokal harus menjadi sebuah kontribusi bagi pengelolaan kawasan konservasi.

4. Pemerintah dan pengelola kawasan konservasi harus bisa bekerjasama dan penguasaan lahan masyarakat lokal dan penggunaan sumberdaya serta sistem kontrol merupakan sebuah pendekatan konservasi keanekaragaman hayati

Perubahan struktural dan revolusi pengelolaan kawasan konservasi harus segera dilakukan, agar pengelola tidak sekedar menjadi *administrator* ijin bagi peneliti asing, atau sekedar menjadi “penjaga kawasan” (kerjaan ini pun juga tidak dilakukan, kecuali patroli), tetapi menjadi garda depan penelitian daerah kawasan konservasi, menjaga dan memanfaatkan kawasan, untuk kepentingan-kepentingan jangka panjang sesuai tujuan yang telah ditetapkan.

## **2.2 Kawasan Konservasi Bengkulu Beserta Ekosistemnya**

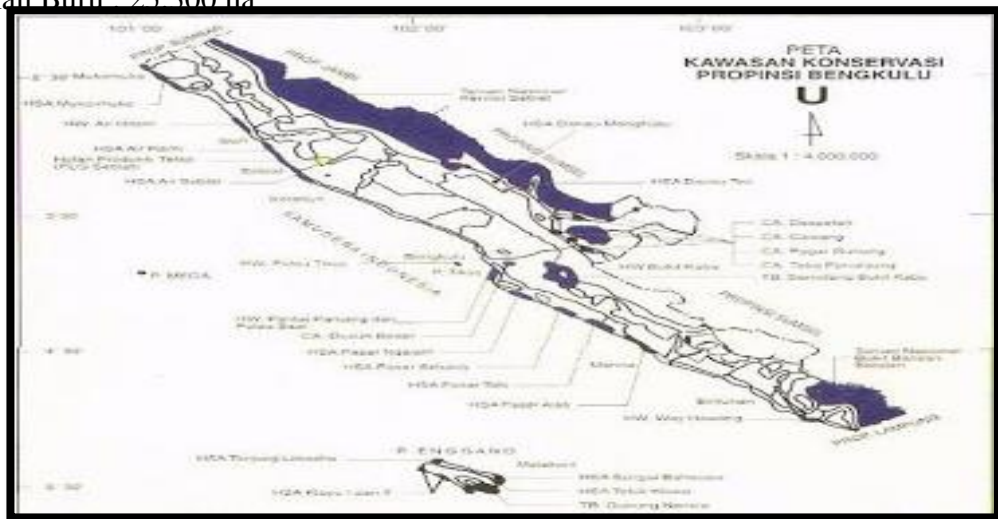
Di Indonesia, berdasarkan peraturan perundang-undangan, Konservasi adalah pengelolaan sumber daya alam hayati yang pemanfaatannya dilakukan secara bijaksana untuk menjamin kesinambungan persediaannya dengan tetap memelihara dan meningkatkan kualitas keanekaragaman dan nilainya. Cagar alam dan suaka margasatwa merupakan Kawasan Suaka Alam (KSA), sementara taman nasional, taman hutan raya, dan taman wisata alam merupakan Kawasan Pelestarian Alam (KPA). Kawasan konservasi dalam arti yang luas, yaitu kawasan dimana konservasi sumber daya alam hayati dilakukan. Di dalam peraturan perundang-undangan Indonesia yang ada, tidak memuat definisi mengenai kawasan konservasi secara jelas. Adapun pengertian kawasan konservasi yang ditemukan dan digunakan oleh Direktorat Jenderal Perlindungan Hutan dan Konservasi Alam (PHKA), Departemen Kehutanan adalah kawasan yang ditetapkan sebagai kawasan suaka alam, kawasan pelestarian alam, taman buru dan hutan lindung. Sementara itu istilah-istilah yang lebih dikenal adalah “kawasan lindung”.

Provinsi Bengkulu terletak di antara 2° 17” - 3 ° 31” Lintang Selatan dan 101 ° 01” - 103° 41” Bujur Timur. Luas wilayah lebih kurang 1.978.870 hektar atau 19.788,7 km<sup>2</sup>, Kawasan konservasi di Provinsi Bengkulu seluruhnya seluas 437.317,95 ha terdiri dari Cagar Alam seluas 15.541,15 ha, Taman nasional seluas 380.064 ha, Taman Wisata Alam seluas 15.290,80 ha, Tahura seluas 1.122 ha, dan Taman Buru seluas 25.300 ha. Kawasan hutan dan perairan Provinsi Bengkulu



berdasarkan keputusan Menteri Kehutanan No. 420/Kpts-II/1999 tanggal 15/06/1999 seluas 920.964 ha, terdiri dari:

1. Luas KPA dan KSA : 419.582 ha
2. Hutan Lindung : 252.042 ha
3. Hutan Produksi Terbatas : 189.075 ha
4. Hutan Produksi : 34.965 ha
5. Hutan Produksi yg dapat dikonversi : -
6. Taman Buru : 25 300 ha



Gambar 2.1 Peta Kawasan Konservasi Provinsi Bengkulu

(Sumber: <http://putraraafflesia.wordpress.com/2009/02/09/pariwisata-bengkulu>)



Gambar 2.2 Peta Kawasan Hutan Provinsi Bengkulu

(Sumber : Dishut Provinsi Bengkulu)

Provinsi Bengkulu memiliki kekayaan flora unik seperti *Rafflesia arnoldy* yang ditemukan pertama kali oleh Sir Thomas Raffles dan Dr. Arnoldy di Dusun Lubuk Tapi tahun 1818. Bunga ini adalah bunga terbesar di dunia berdiameter 100 cm. Bunga ini membutuhkan 6 sampai 8 bulan untuk tumbuh dan 15 hari setelah itu untuk berbunga. Keunikan dari bunga ini adalah tidak memiliki akar, daun dan batang. Tumbuhan ini termasuk parasit karena memiliki klorofil dan haustoria yang mengeluarkan bau cukup menyegat busuk untuk menarik serangga. Bunga ini sering tumbuh dan ditemukan di Taba Penanjung I dan Taba Penanjung III sekitar Bengkulu Tengah, daerah di wilayah kabupaten Kepahiang, dan daerah di wilayah kabupaten Rejang Lebong.

Ada juga bunga Kibut (*Amorphopalus titanuum*). Bunga ini sangat menarik dan cantik karena tidak memiliki batang tetapi memiliki bunga yang tingginya mencapai 3 m dan kuat tumbuh di atas tanah. Bunga ini tumbuh di sekitar Rejang Lebong mengelilingi Kepahiang, Bengkulu Utara, dan Bengkulu Selatan. Provinsi Bengkulu juga memiliki beragam tanaman anggrek di antaranya anggrek air atau *Vanda hookeriana*. Anggrek air ini hanya terdapat di Danau Dendam Tak Sudah terletak sekitar 5 km dari kota Bengkulu. Beberapa macam anggrek liar dan alami lainnya dapat Anda temukan di provinsi Bengkulu.

Bengkulu memiliki berbagai macam kekayaan hutan seperti kayu medang, meranti, rattan, dan damar. Sementara tanaman lainnya yang dibudidayakan oleh masyarakatnya adalah minyak sawit, getah karet, kopi, durian, jeruk, dan sayuran. Selain itu, potensi kekayaan fauna juga tersebar luas di Provinsi Bengkulu. Fauna yang ada di Bengkulu beraneka ragam seperti macan, kijang, gajah, monyet, dan rangkong. Ada juga tempat latihan hewan gajah yaitu di Way Kambas *Elephant Training Center* (ETC) di Seblat yang terletak di sebelah sungai Seblat, Putri Hijau, Bengkulu utara. Tempat latihan ini adalah salah satu dari tempat latihan yang ada di Indonesia (ETC lainnya ada di Lhokseumawe, Aceh Sebangau, Riau dan Sebokor, Sumatera Selatan). Pusat pelatihan tersebut terletak 132 km dari Bengkulu atau sekitar 3 jam perjalanan.

Provinsi Bengkulu juga memiliki taman laut sekitar Pulau Enggano, Taman Nasional Kerinci Seblat (TNKS), Taman Berburu di Gunung Nanu'ua, dimana hutannya yang masih alami yang terletak di pulau Enggano. Hutan tersebut

dijadikan area berburu banteng liar, bore (babi liar), kijang, monyet, dan beberapa jenis hewan lainnya.

## **2.3 Sistem Informasi Geografis**

### **2.3.1 Konsep Dasar Sistem**

Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran yang tertentu (Andi Kristanto, 2007). Kata sistem mempunyai beberapa pengertian, tergantung dari sudut pandang mana kata tersebut didefinisikan. Secara garis besar ada dua kelompok pendekatan sistem, yaitu :

1. Pendekatan sistem yang lebih menekankan pada elemen-elemen atau kelompoknya didefinisikan sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu aturan tertentu.
2. Pendekatan sistem sebagai jaringan kerja dari prosedur, yang lebih menekankan urutan operasi didalam sistem. didefinisikan sebagai urutan operasi kerja (tulis-menulis), yang biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi bisnis yang terjadi.
3. Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen sistem (*component*), batasan sistem (*boundary*), lingkungan luar sistem (*environment*), penghubung sistem (*interface*), masukan sistem (*input*), keluaran sistem (*output*), pengolahan sistem (*process*) dan sasaran (*objective*) atau tujuan (*goal*).

### **2.3.2 Konsep Dasar Informasi**

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya (Jogiyanto, 2005). Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal datum atau item-item. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan

kesatuan nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat yang tertentu. Informasi yang berkualitas memiliki 3 kriteria, yaitu :

1. Akurat (*accurate*)

Informasi harus bebas dari kesalahan, tidak bias ataupun menyesatkan. Akurat juga berarti bahwa informasi itu harus dapat dengan jelas mencerminkan maksudnya.

2. Tepat pada waktunya (*timeliness*)

Informasi yang datang pada penerima tidak boleh terlambat. Di dalam pengambilan keputusan, informasi yang sudah usang tidak lagi bernilai. Bila informasi datang terlambat sehingga pengambilan keputusan terlambat dilakukan, hal itu dapat berakibat fatal bagi perusahaan.

3. Relevan (*relevance*)

Informasi yang disampaikan harus mempunyai keterkaitan dengan masalah yang akan dibahas dengan informasi tersebut. Informasi harus bermanfaat bagi pemakainya. Di samping karakteristik, nilai informasi (*value of information*) ditentukan oleh dua hal, yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih besar dibanding biaya untuk mendapatkannya.

### **2.3.3 Konsep Dasar Sistem Informasi**

Sistem informasi adalah suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi (al-bahra bil ladjamudin, 2002). Sistem informasi disudut pandang yang berbeda merupakan suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan. Sistem informasi dalam suatu organisasi dapat dikatakan sebagai suatu sistem yang menyediakan informasi bagi semua tingkatan dalam organisasi tersebut kapan saja diperlukan. Sistem ini menyimpan, mengambil, mengubah, mengolah dan

mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi atau peralatan sistem lainnya.

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (*building blok*), yang terdiri dari komponen input, komponen model, komponen output, komponen teknologi, komponen *hardware*, komponen *software*, komponen basis data, dan komponen kontrol. Semua komponen tersebut saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran.

1. Komponen input

Input mewakili data yang masuk kedalam sistem informasi. Input disini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Komponen model

Komponen ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Komponen output

Hasil dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.

4. Komponen teknologi

Teknologi merupakan *toolbox* dalam sistem informasi, teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan.

5. Komponen *hardware*

*Hardware* berperan penting sebagai suatu media penyimpanan vital bagi sistem informasi yang berfungsi sebagai tempat untuk menampung database atau lebih mudah dikatakan sebagai sumber data dan informasi untuk memperlancar dan mempermudah kerja dari sistem informasi.

6. Komponen *software*

Software berfungsi sebagai tempat untuk mengolah, menghitung dan memanipulasi data yang diambil dari *hardware* untuk menciptakan suatu informasi.

7. Komponen basis data

Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi menggunakan perangkat lunak paket yang disebut DBMS (*Database Management System*).

8. Komponen kontrol

Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperatur, air, debu, kecurangan, kegagalan sistem itu sendiri, ketidak efisienan, sabotase dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

### **2.3.4 Sejarah Pengembangan Sistem Informasi Geografis**

Tiga puluh lima ribu tahun yang lalu, di dinding gua Lascaux, Perancis, para pemburu Cro-Magnon menggambar hewan mangsa mereka, juga garis yang dipercaya sebagai rute migrasi hewan-hewan tersebut. Catatan awal ini sejalan dengan dua elemen struktur pada sistem informasi geografis modern sekarang ini, arsip grafis yang terhubung ke database atribut. Pada tahun 1700-an teknik survei modern untuk pemetaan topografis diterapkan, termasuk juga versi awal pemetaan tematis, misalnya untuk keilmuan atau data sensus. Awal abad ke-20 memperlihatkan pengembangan "litografi foto" dimana peta dipisahkan menjadi beberapa lapisan (*layer*). Perkembangan perangkat keras komputer yang dipacu

oleh penelitian senjata nuklir membawa aplikasi pemetaan menjadi multifungsi pada awal tahun 1960-an. Tahun 1967 merupakan awal pengembangan SIG yang bisa diterapkan di Ottawa, Ontario oleh Departemen Energi, Pertambangan dan Sumber Daya. Dikembangkan oleh Roger Tomlinson, yang kemudian disebut CGIS (*Canadian GIS*), digunakan untuk menyimpan, menganalisis dan mengolah data yang dikumpulkan untuk Inventarisasi Tanah Kanada (CLI - *Canadian land Inventory*) sebuah inisiatif untuk mengetahui kemampuan lahan di wilayah pedesaan Kanada dengan memetakan berbagai informasi pada tanah, pertanian, pariwisata, alam bebas, unggas dan penggunaan tanah pada skala 1:250000. Faktor pemeringkatan klasifikasi juga diterapkan untuk keperluan analisis.

CGIS merupakan sistem pertama di dunia dan hasil dari perbaikan aplikasi pemetaan yang memiliki kemampuan tumpang susun (*overlay*), penghitungan, pendijitalan/pemindaian (*digitizing/scanning*), mendukung sistem koordinat nasional yang membentang di atas benua Amerika, memasukkan garis sebagai arc yang memiliki topologi dan menyimpan atribut dan informasi lokasional pada berkas terpisah. Pengembangnya, seorang geografer bernama Roger Tomlinson kemudian disebut "Bapak SIG". CGIS bertahan sampai tahun 1970-an dan memakan waktu lama untuk penyempurnaan setelah pengembangan awal, dan tidak bisa bersaing dengan aplikasi pemetaan komersil yang dikeluarkan beberapa *vendor* seperti Intergraph.

Perkembangan perangkat keras mikro komputer memacu vendor lain seperti ESRI dan CARIS berhasil membuat banyak fitur SIG, menggabungkan pendekatan generasi pertama pada pemisahan informasi spasial dan atributnya, dengan pendekatan generasi kedua pada organisasi data atribut menjadi struktur database. Perkembangan industri pada tahun 1980-an dan 1990-an memacu lagi pertumbuhan SIG pada *workstation* UNIX dan komputer pribadi. Pada akhir abad ke-20, pertumbuhan yang cepat di berbagai sistem dikonsolidasikan dan distandarisasikan menjadi platform lebih sedikit, dan para pengguna mulai mengeksport menampilkan data SIG lewat internet, yang membutuhkan standar pada format data dan transfer. Indonesia sudah mengadopsi sistem ini sejak Pelita ke-2 ketika LIPI mengundang UNESCO dalam menyusun "Kebijakan dan Program Pembangunan Lima Tahun Tahap Kedua (1974-1979)" dalam

pembangunan ilmu pengetahuan, teknologi dan riset. Pada masa ini telah dikembangkan sistem-sistem yang secara khusus dibuat untuk menangani masalah informasi yang bereferansi geografis dalam berbagai cara dan bentuk. Masalah-masalah ini mencakup:

1. Pengorganisasian data dan informasi
2. Penempatan informasi pada lokasi tertentu.
3. Melakukan komputasi, memberikan ilusi keterhubungan satu sama lainnya (koneksi), beserta analisa-analisa spasial lainnya.

Sebutan umum untuk sistem-sistem yang menangani masalah-masalah tersebut adalah Sistem Informasi Geografis. Dalam literatur, Sistem Informasi Geografis dipandang sebagai hasil perpaduan antara sistem komputer untuk bidang Kartografi (CAC) atau sistem komputer untuk bidang perancangan (CAD) dengan teknologi basis data (database). Pada awalnya, data geografis hanya disajikan di atas peta dengan menggunakan symbol, garis dan warna. Elemen-elemen geografis ini dideskripsikan di dalam legendanya misalnya: garis hitam tebal untuk jalan utama, garis hitam tipis untuk jalan sekunder dan jalan-jalan yang berikutnya. Selain itu, berbagai data yang dioverlay berdasarkan sistem koordinat yang sama. Akibatnya sebuah peta menjadi media yang efektif baik sebagai alat presentasi maupun sebagai bank tempat penyimpanan data geografis. Tetapi media peta masih mengandung kelemahan atau keterbatasan. Informasi-informasi yang disimpan, diproses dan dipresentasikan dengan suatu cara tertentu, dan biasanya untuk tujuan tertentu pula, tidak mudah untuk merubah presentasi tersebut karena peta selalu menyediakan gambar atau simbol unsur geografis dengan bentuk yang tetap walaupun diperlukan untuk kebutuhan yang berbeda.

### **2.3.5 Sistem Informasi Geografis (SIG)**

SIG adalah sistem informasi yang didasarkan pada kerja komputer yang memasukkan, mengelola, memanipulasi dan menganalisa data serta memberi uraian (Aronaff, 1989). Sistem Informasi Geografi (SIG) atau *Geographic Information System* (SIG) merupakan suatu sistem informasi yang dirancang untuk bekerja dengan data yang bereferensi spasial atau berkoordinat geografi atau dengan kata lain suatu SIG adalah suatu sistem basis data dengan



kemampuan khusus untuk menangani data yang bereferensi keruangan (spasial) bersamaan dengan seperangkat operasi kerja. Sistem Informasi Geografis dibagi menjadi dua kelompok yaitu sistem manual (analog), dan sistem otomatis (yang berbasis digital komputer). Perbedaan yang paling mendasar terletak pada cara pengelolaannya. Sistem Informasi manual biasanya menggabungkan beberapa data seperti peta, lembar transparansi untuk tumpang susun (*overlay*), foto udara, laporan statistik dan laporan survey lapangan. Kesemua data tersebut dikompilasi dan dianalisis secara manual dengan alat tanpa komputer. Sedangkan Sistem Informasi Geografis otomatis telah menggunakan komputer sebagai sistem pengolah data melalui proses digitasi. Sumber data digital dapat berupa citra satelit atau foto udara digital serta foto udara yang terdigitasi. Data lain dapat berupa peta dasar terdigitasi.

Pengertian SIG/SIG saat ini lebih sering diterapkan bagi teknologi informasi spasial atau geografi yang berorientasi pada penggunaan teknologi komputer. Dalam hubungannya dengan teknologi komputer, Arronoff (1989) mendefinisikan SIG sebagai sistem berbasis komputer yang memiliki kemampuan dalam menangani data bereferensi geografi yaitu pemasukan data, manajemen data (penyimpanan dan pemanggilan kembali), memanipulasi dan analisis data, serta keluaran sebagai hasil akhir (*output*). Sedangkan Burrough (1986) mendefinisikan Sistem Informasi Geografis (SIG) sebagai sistem berbasis komputer yang digunakan untuk memasukkan, menyimpan, mengelola, menganalisis dan mengaktifkan kembali data yang mempunyai referensi keruangan untuk berbagai tujuan yang berkaitan dengan pemetaan dan perencanaan. Komponen utama Sistem Informasi Geografis dapat dibagi kedalam 4 komponen utama yaitu: perangkat keras (*digitizer, scanner, Central Processing Unit* (CPU), *hard-disk*, dan lain-lain), perangkat lunak (ArcView, Idrisi, ARC/INFO, ILWIS, MapInfo, dan lain-lain), organisasi (manajemen) dan pemakai (*user*). Kombinasi yang benar antara keempat komponen utama ini akan menentukan kesuksesan suatu proyek pengembangan Sistem Informasi Geografis.

Aplikasi SIG dapat digunakan untuk berbagai kepentingan selama data yang diolah memiliki referensi geografi, maksudnya data tersebut terdiri dari fenomena atau objek yang dapat disajikan dalam bentuk fisik serta memiliki

lokasi keruangan. Tujuan pokok dari pemanfaatan Sistem Informasi Geografis adalah untuk mempermudah mendapatkan informasi yang telah diolah dan tersimpan sebagai atribut suatu lokasi atau obyek. Ciri utama data yang bisa dimanfaatkan dalam Sistem Informasi Geografis adalah data yang telah terikat dengan lokasi dan merupakan data dasar yang belum dispesifikasi. Data-data yang diolah dalam SIG pada dasarnya terdiri dari data spasial dan data atribut dalam bentuk digital, dengan demikian analisis yang dapat digunakan adalah analisis spasial dan analisis atribut. Data spasial merupakan data yang berkaitan dengan lokasi keruangan yang umumnya berbentuk peta. Sedangkan data atribut merupakan data tabel yang berfungsi menjelaskan keberadaan berbagai objek sebagai data spasial.

Penyajian data spasial mempunyai tiga cara dasar yaitu dalam bentuk titik, bentuk garis dan bentuk area (*polygon*). Titik merupakan kenampakan tunggal dari sepasang koordinat x,y yang menunjukkan lokasi suatu obyek berupa ketinggian, lokasi kota, lokasi pengambilan sample dan lain-lain. Garis merupakan sekumpulan titik-titik yang membentuk suatu kenampakan memanjang seperti sungai, jalan, kontur dan lain-lain. Sedangkan area adalah kenampakan yang dibatasi oleh suatu garis yang membentuk suatu ruang homogen, misalnya: batas daerah, batas penggunaan lahan, pulau dan lain sebagainya. Struktur data spasial dibagi dua yaitu model data raster dan model data vektor. Data raster adalah data yang disimpan dalam bentuk kotak segi empat (*grid/sel*) sehingga terbentuk suatu ruang yang teratur. Data vektor adalah data yang direkam dalam bentuk koordinat titik yang menampilkan, menempatkan dan menyimpan data spasial dengan menggunakan titik, garis atau area (*polygon*).

Sistem informasi geografi menyajikan informasi keruangan beserta atributnya yang terdiri dari beberapa komponen utama yaitu:

1. Masukan data merupakan proses pemasukan data pada komputer dari peta (peta topografi dan peta tematik), data statistik, data hasil analisis penginderaan jauh data hasil pengolahan citra digital penginderaan jauh, dan lain-lain. Data-data spasial dan atribut baik dalam bentuk analog maupun data digital tersebut dikonversikan kedalam format yang diminta oleh perangkat lunak sehingga terbentuk basisdata (*database*).

2. Penyimpanan data dan pemanggilan kembali (*data storage* dan *retrieval*) ialah penyimpanan data pada komputer dan pemanggilan kembali dengan cepat (penampilan pada layar monitor dan dapat ditampilkan/cetak pada kertas).
3. Manipulasi data dan analisis ialah kegiatan yang dapat dilakukan berbagai macam perintah misalnya *overlay* antara dua tema peta, membuat *buffer zone* jarak tertentu dari suatu area atau titik dan sebagainya.
4. Pelaporan data ialah dapat menyajikan data dasar, data hasil pengolahan data dari model menjadi bentuk peta atau data tabular.

Alasan mengapa perlu menggunakan SIG, diantaranya adalah:

1. SIG menggunakan data spasial maupun atribut secara terintegrasi.
2. SIG dapat digunakan sebagai alat bantu interaktif yang menarik dalam usaha meningkatkan pemahaman mengenai konsep lokasi, ruang, kependudukan, dan unsur-unsur geografi yang ada dipermukaan bumi.
3. SIG dapat memisahkan antara bentuk presentasi dan basis data.
4. SIG memiliki kemampuan menguraikan unsur-unsur yang ada dipermukaan bumi kedalam beberapa layer atau *coverage* data spasial.
5. SIG memiliki kemampuan yang sangat baik dalam memvisualisasikan data spasial berikut atributnya.
6. Semua operasi SIG dapat dilakukan secara interaktif.
7. SIG dengan mudah menghasilkan peta-peta tematik.
8. Semua operasi SIG dapat di *customize* dengan menggunakan perintah-perintah dalam bahasa *script*.
9. Perangkat lunak SIG menyediakan fasilitas untuk berkomunikasi dengan perangkat lunak lain.
10. SIG sangat membantu pekerjaan yang erat kaitannya dengan bidang spasial dan geoinformatika.

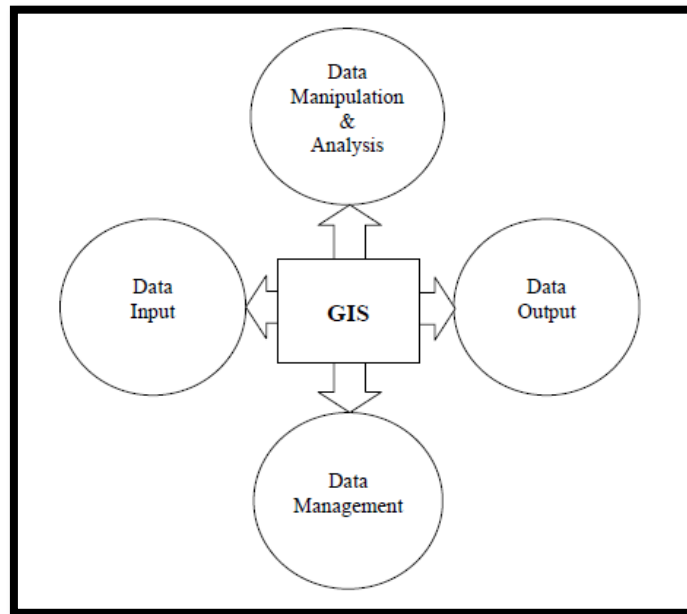
SIG adalah alat yang handal untuk menangani data spasial, dimana dalam SIG data dipelihara dalam bentuk digital sehingga data ini lebih padat dibanding dalam bentuk peta cetak, tabel atau dalam bentuk konvensional lainnya yang akhirnya akan mempercepat pekerjaan dan meringankan biaya yang diperlukan. Sarana utama untuk penanganan data spasial adalah SIG. SIG didesain untuk

menerima data spasial dalam jumlah besar dari berbagai sumber dan mengintegrasikannya menjadi sebuah informasi, salah satu jenis data ini adalah data penginderaan jauh. Penginderaan jauh mempunyai kemampuan menghasilkan data spasial yang susunan geometrinya mendekati keadaan sebenarnya dengan cepat dan dalam jumlah besar. SIG akan memberi nilai tambah pada kemampuan penginderaan jauh dalam menghasilkan data spasial yang besar dimana pemanfaatan data penginderaan jauh tersebut tergantung pada cara penanganan dan pengolahan data yang akan mengubahnya menjadi informasi yang berguna.

### **2.3.6 Subsistem Sistem Informasi Geografis**

Sistem Informasi Geografis dapat diuraikan menjadi beberapa subsistem sebagai berikut:

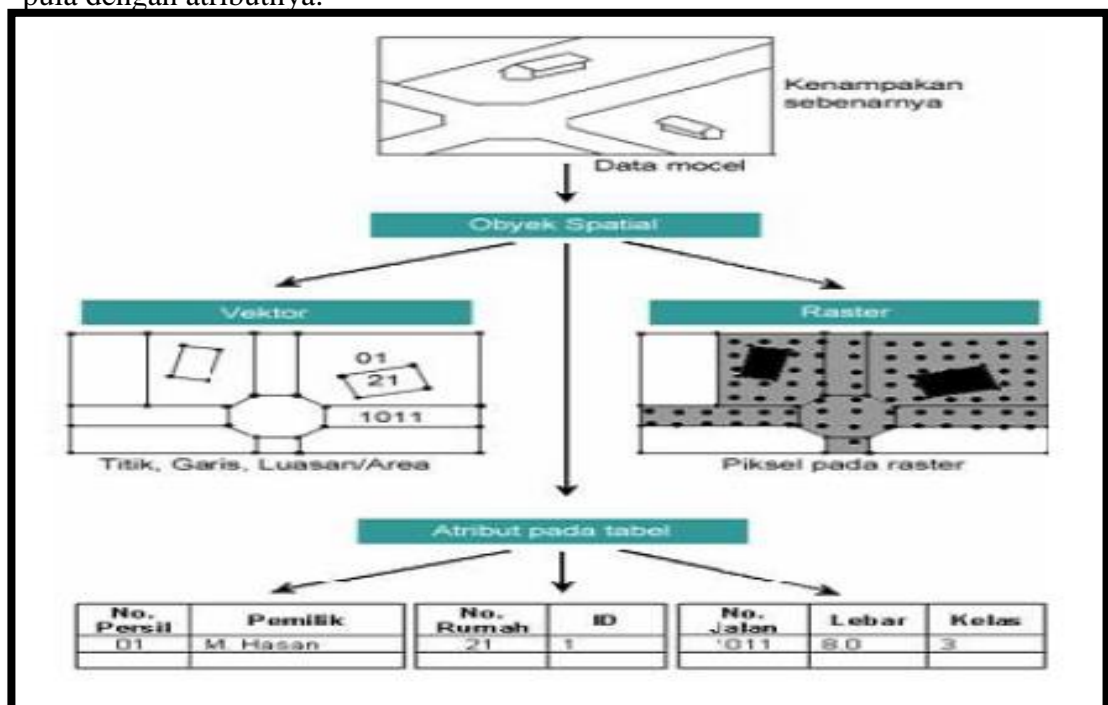
1. *Data Input*: Subsistem ini bertugas untuk mengumpulkan data dan mempersiapkan data spasial dan atribut dari berbagai sumber dan bertanggung jawab dalam mengkonversi atau mentransfortasikan formatformat data-data aslinya kedalam format yang dapat digunakan oleh SIG.
2. *Data output*: Subsistem ini menampilkan atau menghasilkan keluaran seluruh atau sebagian basis data baik dalam bentuk softcopy maupun bentuk hardcopy seperti: tabel, grafik dan peta.
3. *Data Management*: Subsistem ini mengorganisasikan baik data spasial maupun data atribut ke dalam sebuah basis data sedemikian rupa sehingga mudah dipanggil, diupdate dan diedit.
4. *Data Manipulation & Analysis*: Subsistem ini menentukan informasiinformasi yang dapat dihasilkan oleh SIG dan melakukan manipulasi serta pemodelan data untuk menghasilkan informasi yang diharapkan, dapat di lihat pada Gambar 2.1.



Gambar 2.3 Subsistem SIG (Sumber: <http://arashirin.wordpress.com>)

### 2.3.7 Konsep Model Data Spasial

Data spasial merupakan data yang paling penting dalam SIG. Seperti penjelasan diatas data spasial ada 2 macam yaitu data raster dan data vektor. Dibawah ini adalah salah satu contoh konsep data geospasial/spasial dihubungkan pula dengan atributnya.



Gambar 2.4 Contoh Gambar Geospasial

(Sumber: <http://bebasbanjir2025.wordpress.com/>)

### **2.3.8 Data Vektor**

Model data vektor yang menampilkan, menempatkan dan menyimpan data spasial dengan menggunakan titik-titik, garis-garis, atau kurva atau polygon beserta atribut-atributnya. Bentuk dasar representasi data spasial didalam sistem model data vektor, didefinisikan oleh sistem koordinat kartesian dua dimensi (x,y).

### **2.3.9 Data Raster**

Model data raster menampilkan, menempatkan dan menyimpan spasial dengan menggunakan struktur matriks atau pixel-pixel yang membentuk grid. Akurasi model data ini sangat bergantung pada resolusi atau ukuran pixelnya (*sel grid*) di permukaan bumi. Contoh data raster adalah citra satelit misalnya Spot, Landsat, dll. Konsep model data ini adalah dengan memberikan nilai yang berbeda untuk tiap-tiap pixel atau grid dari kondisi yang berbeda.

### **2.3.10 Kebutuhan Terhadap Sistem Informasi Geografis**

Berikut ini beberapa alasan mengapa SIG dibutuhkan:

1. Penangan data geospasial sangat buruk.
2. Peta dan statistik sangat cepat kadaluarsa.
3. Data dan informasi sering tidak akurat.
4. Tidak ada pelayanan penyediaan data.
5. Tidak ada pertukaran data.

Dengan SIG diterapkan, didapat keuntungan berikut:

1. Penanganan data geospasial menjadi lebih baik dalam format baku.
2. Devisi dan pemutakhiran data menjadi lebih mudah.
3. Data geospasial dan informasi lebih mudah dicari, dianalisis dan direpresentasikan.
4. Menjadi produk bernilai tambah.
5. Data geospasial dapat dipertukarkan.
6. Produktivitas staf meningkat dan lebih efisien

7. Penghematan waktu dan biaya.
8. Keputusan yang akan diambil menjadi lebih baik

Kelebihan-kelebihan SIG dapat dilihat pada Tabel 2.1.

Peta	GIS	Manual
Penyimpanan	Database Digital Baku Dan Terpadu	Skala dan standar berbeda
Pemanggilan Kembali	Pencatatan dengan Komputer	Cek Manual
Pemutakhiran	Sistematis	Mahal dan memakan waktu
Analisis Overlay	Sangat cepat	Memakan waktu dan tenaga
Penayangan	Murah dan Cepat	Mahal

### 2.3.11 Sistem Komputer Untuk SIG

Sistem komputer biasanya terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*).

#### 1. Perangkat Keras

Terdiri dari beberapa komponen sebagai berikut :

##### 1) *Central processing unit* (CPU)

CPU menjalankan program komputer dan mengendalikan operasi seluruh komponen. Biasanya digunakan CPU untuk komputer pribadi (PC/*personal computer*), atau *work station* pada sebuah jaringan komputer.

##### 2) *Memory* (Memori)

Memori Utama adalah bagian paling esensi pada komputer seluruh data dan program berada pada memori utama untuk akses yang lebih cepat. Dibutuhkan setidaknya memori berkapasitas 64 MB untuk SIG berbasis PC. Memori Tambahan digunakan data berukuran besar baik permanen maupun semi-permanen, dengan akses lebih rendah dibanding memori utama. Dikenal juga sebagai media penyimpanan data, seperti *hard disk*, disket (*floppy disk*), pita magnetis atau cakram padat optis (*CD-ROM*). Untuk harddisk dibutuhkan setidaknya yang berkapasitas 1 GB.

### 3) Alat Tambahan (*Peripherals*)

Alat Masukan (*Input Devices*) meliputi *key board, mouse, digitizers*, pemindai (*scanner*), kamera digital, *workstation fotogrametris digital*. Alat Keluaran (*Output devices*) meliputi monitor berwarna, *printer, plotter* berwarna, perekam film, dll.

## 2. Perangkat Lunak

Terdiri atas sistem operasi, compiler dan program aplikasi. Sistem Operasi (*Operating System*) mengendalikan seluruh operasi program, juga menghubungkan perangkat keras dengan program aplikasi. Untuk PC meliputi MS-DOS (IBM PCs) dan Windows adalah sistem operasi yang banyak digunakan. Untuk *Workstation*: UNIX dan VMS adalah OS yang dominan. *Compiler* menerjemahkan program yang ditulis dalam bahasa komputer pada kode mesin sehingga CPU mampu menjalankan program yang harus dieksekusi. Bahasa kompilasi yang biasa digunakan adalah C, Pascal, Fortran, Basic, dll.

### 2.3.12 SIG Sebagai Ilmu Multi Disiplin

SIG sebagai ilmu multi disiplin terpadu terdiri atas beberapa disiplin ilmu berikut:

- |                      |                           |
|----------------------|---------------------------|
| a. Geografi          | g. Statistika             |
| b. Kartografi        | h. Operations Research    |
| c. Penginderaan Jauh | i. Ilmu Komputer          |
| d. Fotogrammetri     | j. Matematika             |
| e. Ilmu Ukur Tanah   | k. Perencanaan Sipil      |
| f. Geodesi           | l. Perencanaan Kota, dll. |



SIG memiliki banyak nama alternatif yang sudah digunakan bertahun-tahun menurut cakupan aplikasi dan bidang khusus masing-masing, sebagai berikut:

1. Sistem Informasi Lahan (*Land Information System - LIS*).
2. Pemetaan terautomatisasi dan Pengelolaan Fasilitas (*AM/FM-Automated Mapping and Facilities Management*).
3. Sistem Informasi Lingkungan (*Environmental Information System - EIS*).
4. Sistem Informasi Sumber Daya (*Resources Information System*).
5. Sistem Informasi Perencanaan (*Planning Information System*).
6. Sistem Penanganan Data keruangan (*Spatial Data Handling System*).

SIG kini menjadi disiplin ilmu yang independen dengan nama "Geomatic", "Geoinformatics", atau "*Geospatial Information Science*" yang digunakan pada berbagai departemen pemerintahan dan universitas.

### **2.3.13 Cakupan Aplikasi SIG**

Cakupan utama Aplikasi SIG dapat dikelompokkan ke dalam lima kategori.

1. Pengelolaan Fasilitas

Peta skala besar dan akurat, dan analisis jaringan (*network analysis*) digunakan untuk pengelolaan utilitas kota. AM/FM biasanya digunakan pada tujuan ini.

2. Pengelolaan Sumber Daya Alam dan Lingkungan

Untuk tujuan ini digunakan peta skala menengah dan kecil, dan teknik tumpang tindih (*overlay*) digabungkan dengan foto udara dan citra satelit untuk analisis dampak lingkungan dan pengelolaan sumber daya alam.

3. Jaringan Jalan

Untuk fungsi jaringan jalan digunakan peta skala besar dan menengah, dan analisis keruangan yang digunakan untuk rute kendaraan, lokasi perumahan dan jalan, dll.

#### 4. Perencanaan dan Rekayasa

Digunakan peta skala besar dan menengah, dan model rekayasa untuk perencanaan sipil.

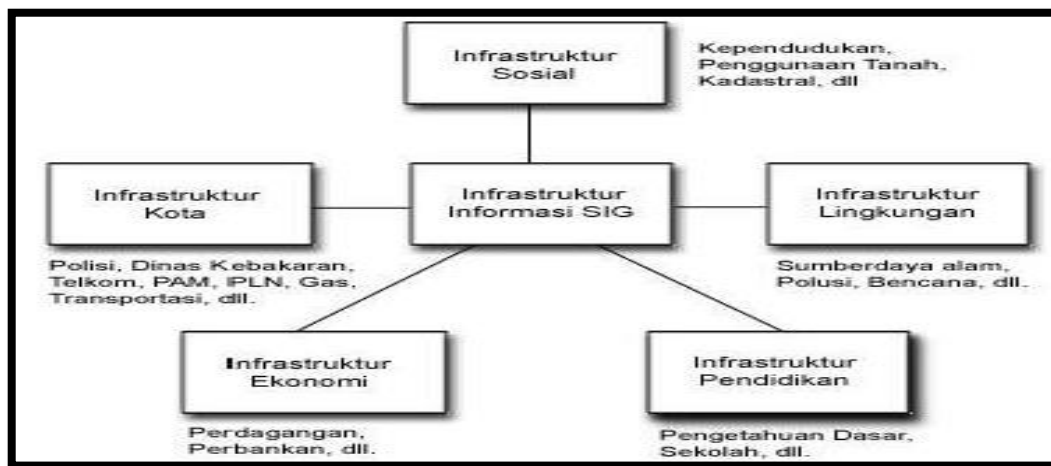
#### 5. Sistem Informasi Lahan

Digunakan peta kadastral skala besar atau peta persil tanah, dan analisis keruangan untuk informasi kadastral, pajak, dll.

### 2.3.14 SIG Sebagai Infrastruktur Informasi

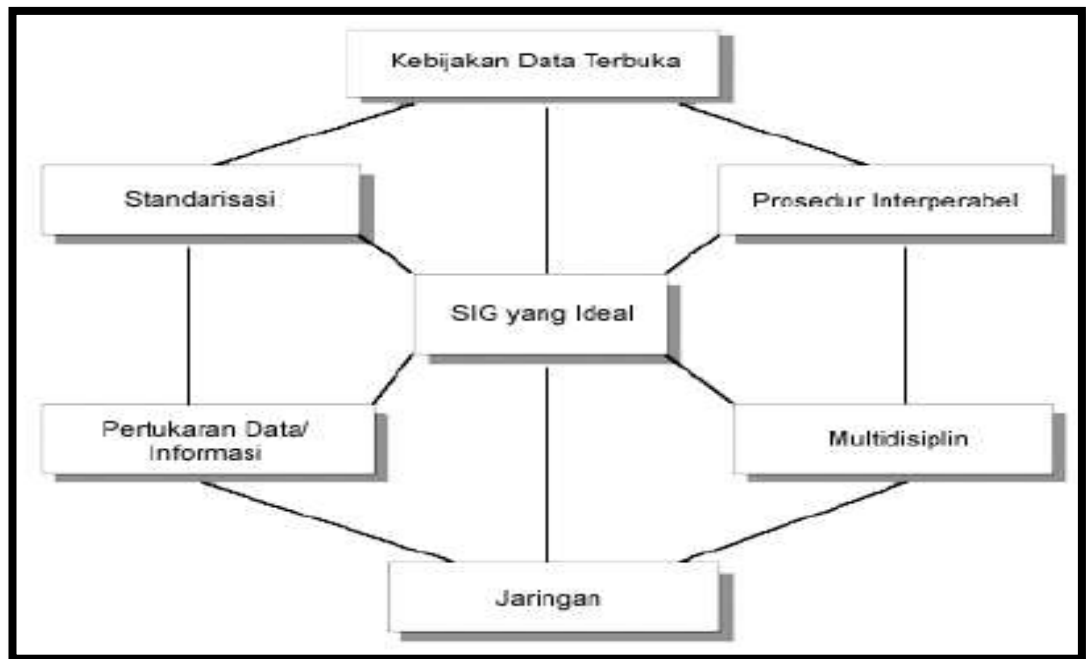
Informasi menjadi isu utama dalam era komputer sekarang ini, karena informasi memberikan kontribusi pada kualitas hidup seperti tertulis di bawah ini:

1. Infrastruktur sosial, masyarakat yang lebih baik.
2. Infrastruktur lingkungan, pengelolaan yang lebih baik.
3. Infrastruktur kota, kehidupan yang lebih baik.
4. Infrastruktur ekonomi, usaha yang lebih baik.
5. Infrastruktur pendidikan, pengetahuan yang lebih baik.



Gambar 2.5 Infrastruktur Informasi SIG (Sumber: <http://elib.unikom.ac.id/>)

Untuk mendapatkan infrastruktur informasi SIG, isu-isu seperti pada Gambar 2.5 harus dipecahkan dan dikembangkan.



Gambar 2.6 SIG yang ideal (Sumber: <http://elib.unikom.ac.id/>)

#### 1. Kebijakan Data Terbuka

Data dan informasi tentang SIG harus bisa diperoleh oleh siapapun tanpa batasan dan gratis atau murah.

#### 2. Standarisasi

Standar untuk struktur dan format data harus dibangun untuk memungkinkan transfer data dan pertukaran data geospasial.

#### 3. Pertukaran Data/Informasi

Untuk penghematan waktu dan biaya dalam digitasi, pertukaran data harus dikembangkan. Untuk mendukung pekerjaan dengan data geospasial, informasi dan pengalaman harus dipertukarkan antar sesama pengguna.

#### 4. Jaringan

Sistem komputer yang tersebar antar instansi harus dihubungkan dengan jaringan untuk peningkatan akses dan pelayanan.

## 5. Pendekatan Multi Disiplin

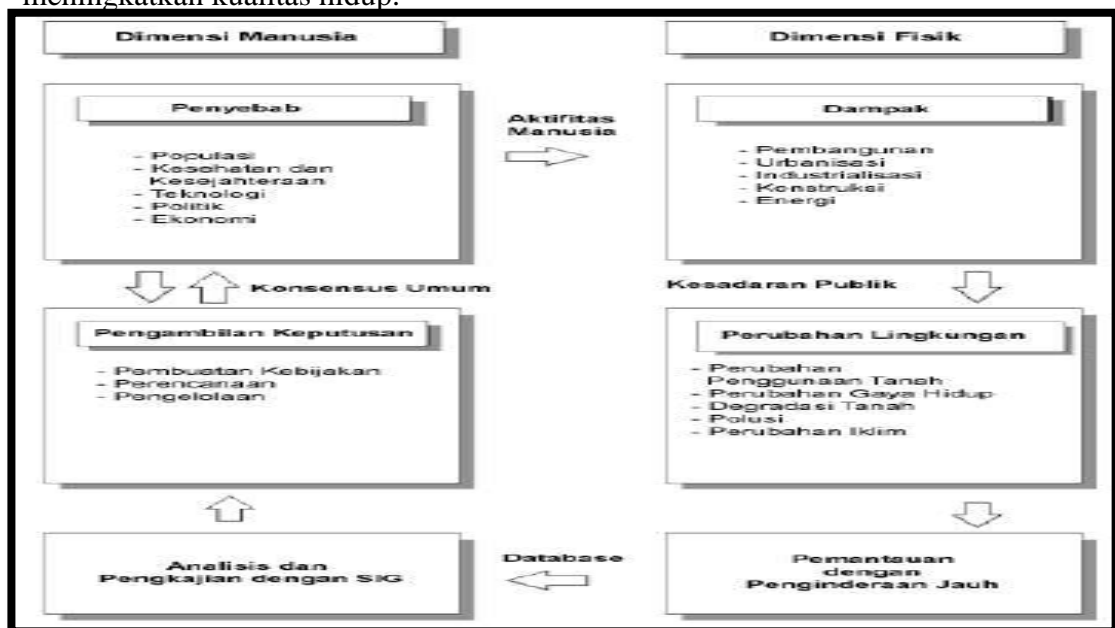
Karena SIG adalah ilmu multi disiplin, maka para ilmuwan, insinyur, teknisi dan tenaga administrasi dari berbagai disiplin harus bekerja sama untuk tujuan bersama.

## 6. Prosedur *Interoperable*

SIG harus dapat dihubungkan dengan prosedur komputer lain seperti CAD, komputer grafis, DEM, dll.

### 2.3.15 SIG Untuk Pengambilan Keputusan

SIG bisa menjadi alat yang sangat penting pada pengambilan keputusan untuk pembangunan berkelanjutan, karena SIG memberikan informasi pada pengambil keputusan untuk analisis dan penerapan database keruangan. Pengambilan keputusan termasuk pembuatan kebijakan, perencanaan dan pengelolaan dapat diimplementasikan secara langsung dengan pertimbangan faktor-faktor penyebabnya melalui suatu konsesus masyarakat. Faktor penyebab itu bisa berupa pertumbuhan populasi, tingkat kesehatan, tingkat kesejahteraan, teknologi, politik, ekonomi dll. yang kemudian ditentukan target dan tujuan untuk meningkatkan kualitas hidup.



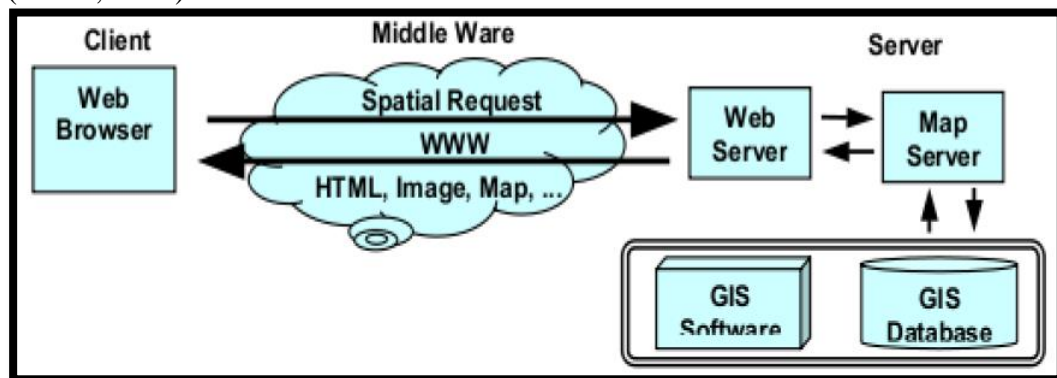
Gambar 2.7 SIG untuk Pengambilan Keputusan

(Sumber: <http://elib.unikom.ac.id/>)

Dua, faktor penyebab dari manusia, elemen kunci dimensi manusia pada pengambilan keputusan, akan memberikan akibat pada lingkungan seperti peningkatan pemakaian sumber daya alam, urbanisasi, industrialisasi, konstruksi, konsumsi energi, dll. Akibat yang terjadi pada manusia ini akan berpengaruh pada perubahan lingkungan, seperti perubahan penggunaan tanah, perubahan gaya hidup, degradasi tanah, polusi, perubahan iklim, dll. Perubahan lingkungan itu dapat dipantau untuk meningkatkan kewaspadaan publik. Penginderaan jauh dapat sangat berguna untuk pemahaman yang lebih baik atas akibat pada manusia dengan perubahan lingkungan, selain penginderaan jauh juga membangun database. Dimensi fisik/lingkungan yang dipantau dengan penginderaan jauh dapat memberikan umpan balik pada manusia melalui analisis dan pengkajian dengan SIG untuk mendukung pengambilan keputusan yang lebih baik. Dalam hal ini, penginderaan jauh harus diintegrasikan dengan SIG.

### 2.3.16 Arsitektur SIG Berbasis Web

Di dalam menjalankan tugas-tugas analisis SIG, SIG berbasis web mirip dengan tipe arsitektur tiga tingkat *client server*. Pemrosesan data geospasial dibagi menjadi bagian yaitu di sisi *server* dan di sisi *client*. Pada bagian *client* biasanya adalah suatu web *browser*/perangkat lain, sedangkan bagian server terdiri dari web *Server*, aplikasi SIG berbasis web dan basis data, seperti terlihat pada gambar 2.8 (Helali, 2001).



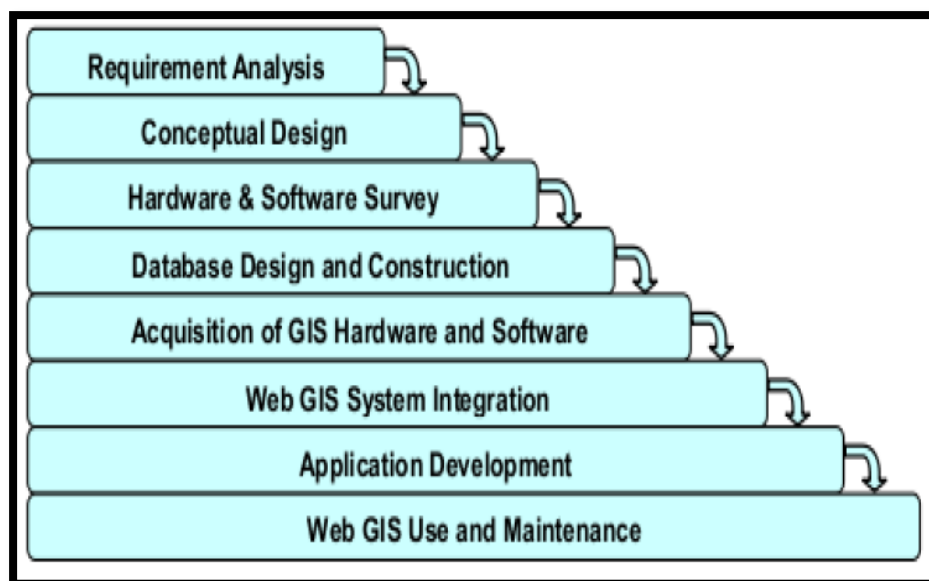
Gambar 2.8 Arsitektur SIG berbasis Web

Model jaringan ini dikenal luas dalam organisasi, dimana beberapa komputer bertindak sebagai server dan lainnya bertindak sebagai *client*. Server mempunyai hak untuk menjalankan aplikasi SIG dan menambahkan sebuah

antarmuka di sisi *client* dan suatu *middleware* di sisi *server* untuk berkomunikasi antara *client* dan aplikasi SIG. Perkembangan terbaru dalam pemrograman berorientasi obyek memungkinkan untuk memproduksi komponen perangkat lunak, dan mengirimnya ke *client* sebelum menjalankannya di mesin *client*, seperti kelas Java, komponen ActiveX dan plug-in. Model ini disebut *thick client* GIS . Pada arsitektur *thick client*, mesin akan melakukan pekerjaan pemrosesan secara lokal. Baik *thin* maupun *thick client system* memiliki beberapa keuntungan dan kelemahan, tetapi mereka bukan merupakan solusi terbaik dalam hal memanfaatkan sumber daya jaringan.

## 2.4 Siklus Pengembangan Perangkat Lunak SIG Berbasis Web

Mengembangkan sebuah SIG berbasis web berbeda dengan sekedar membeli perangkat keras dan perangkat lunak yang sesuai. Beberapa strategi telah diusulkan untuk memberikan keberhasilan dalam implementasinya (Alesheikh-Helali 2001). Strategi implementasi SIG ini telah diuji secara ilmiah dan telah dimodifikasi sehingga kebutuhan suatu pelaksanaan proyek pengembangan SIG berbasis dapat dipenuhi dengan biaya dan waktu yang minimum. Pada gambar 2.9 ditunjukkan siklus pengembangan SIG berbasis web yang digambarkan dalam 8 kegiatan utama, dimulai dengan analisis kebutuhan dan berakhir dengan penggunaan dan pemeliharaan sistem SIG berbasis web (Alesheikh-Helali, 2002).



Gambar 2.9 Siklus pengembangan SIG berbasis web

Penjelasan dari tahapan-tahapan pengembangan SIG berbasis web adalah sebagai berikut :

#### 1. Analisa Kebutuhan (*Requirement Analysis*)

Di dalam tahapan ini harus ditentukan terlebih dahulu tujuan yang ingin dicapai dari sistem yang akan dibuat. Langkah analisis kebutuhan dapat dilakukan melalui wawancara dengan calon pengguna. Langkah ini menghasilkan dua bagian penting dari informasi :

##### 1) Fungsi-fungsi yang diperlukan.

Fungsi-fungsi yang diperlukan adalah fungsi visualisasi dasar seperti Pan, Zoom, dan fungsi lain yang lebih maju seperti identifikasi objek, query spasial, dan jalan terpendek.

##### 2) Data geografis yang diperlukan.

Informasi yang diperoleh dalam kegiatan analisis kebutuhan akan dilanjutkan ketahap berikutnya yaitu disain konseptual SIG

#### 2. Disain konseptual (*Conceptual Design*)

Setelah data yang dibutuhkan telah diidentifikasi, dirancanglah model data yang diwujudkan dalam diagram relasi entitas ( *Entity Relationship Diagram* ).

#### 3. Survey terhadap Ketersediaan Perangkat Keras dan Perangkat Lunak

Pemilihan perangkat lunak yang cocok merupakan langkah penting dalam kesuksesan tahapan implementasi. Perangkat Lunak dievaluasi pada fungsionalitas dan kinerja, dan independen dari *hardware* dan sistem operasi.

#### 4. Perancangan Basis data dan Konstruksi

Tujuan utama pada tahapan ini adalah untuk menentukan "bagaimana" Web GIS merancang kebutuhan aplikasi yang diperlukan. Perancangan basisdata meliputi bagaiman mendefinisikan bagaimana grafis akan dilambangkan (yaitu, warna, berat, ukuran, simbol, dll), bagaimana bentuk struktur *file* grafis, bagaimana bentuk

struktur atribut *file non* grafis, layer mana yang aktif, dalam skala apa layer yang harus ditampilkan, bagaimana produk SIG akan ditampilkan (misalnya, peta tata letak lembar, format laporan, dll), dan manajemen dan pembatasan keamanan apa yang akan dikenakan pada akses file. Kegiatan yang dilakukan adalah sebagai berikut:

- 1) Memilih sumber (dokumen, peta, *file digital*, dll) untuk setiap entitas dan atribut yang termasuk dalam diagram *Entitas-Relationship*.
- 2) Melakukan pengaturan rancangan basisdata (baik secara logik maupun fisik).
- 3) Menentukan prosedur untuk mengkonversi data dari media sumber ke basisdata.
- 4) Menetapkan prosedur untuk mengelola dan memelihara basisdata.
- 5) Akuisisi Perangkat Keras dan Perangkat Lunak SIG

Kegiatan perancangan basisdata dilakukan bersamaan dengan aktifitas pemilihan perangkat keras dan perangkat lunak SIG. Perancangan prosedur dan basisdata tidak dapat diselesaikan sebelum *hardware* dan *software* SIG telah dipilih sementara pada saat yang sama pemilihan hardware dan software SIG tidak dapat diselesaikan sampai SIG yang dipilih dapat menunjukkan fungsi-fungsi yang diperlukan pada data. Berdasarkan kemampuan fungsional, dukungan *vendor*, biaya / *fee* pemeliharaan, dan survei dari perangkat keras yang tersedia / perangkat lunak, yang telah dibeli, dan semua yang diperlukan renovasi ruang, kabel, dan penataan kembali lingkungan hidup telah dilakukan.

#### 6. Integrasi Sistem SIG berbasis Web ( *WebGIS System Integration* )

Pada tahap ini proses pengembangan SIG berbasis web, *hardware* dan *software* telah diakuisisi dan konversi data telah selesai. Tujuan dari tahap ini kemudian untuk mengintegrasikan komponen yang berbeda dari *hardware* dan *software*, untuk menguji dan untuk memastikan seluruh komponen bekerja seperti yang diharapkan, dan untuk memulai semua prosedur yang diperlukan untuk menggunakan SIG.



## 7. Pengembangan Aplikasi (*Application Development*)

Pada tahapan ini seluruh kebutuhan dari software yang telah ditentukan dalam tahapan analisa maupun perancangan diwujudkan dalam bentuk aplikasi. Kemudahan untuk digunakan, *user friendly*, dan *volume data transfer* menjadi hal yang sangat diperhatikan pada tahap ini.

## 8. Penggunaan dan Pemeliharaan SIG berbasis Web

Langkah terakhir dalam implementasi SIG berbasis web adalah untuk menempatkan sistem untuk digunakan. Dengan selesainya sistem integrasi dan pengujian, serta seluruh siap untuk digunakan, maka sistem ini diberikan kepada pengguna. Kegiatan yang dilakukan adalah sebagai berikut :

- 1) Dukungan dan layanan terhadap pengguna di mana aplikasi yang baru akan diterapkan.
- 2) Sistem pemeliharaan (*basisdata, hardware, software*) agar SIG berbasis web berjalan lancar.

## 2.5 WEB Beserta Perangkat Pendukungnya

### 2.5.1 Sejarah Lahirnya WEB

Sejarah Web dimulai pada bulan Maret 1989 ketika Tim Berner-Lee yang bekerja di Laboratorium Fisika Partikel Eropa atau yang dikenal dengan nama *Conseil European pour la Recherche Nuclaire* (CERN) yang berada di Genewa, Swiss, mengajukan protokol atau suatu tatacara untuk berkomunikasi sistem distribusi informasi internet yang digunakan untuk berbagai informasi diantara para fisikawan. Protokol inilah yang selanjutnya dikenal sebagai protokol *World Wide Web* (WWW) dan dikembangkan oleh *World Wide Web Consortium* (W3C). Dimana W3C adalah konsorsium dari sejumlah organisasi yang berkepentingan dalam pengembangan berbagai standar yang berkaitan dengan Web. W3C meletakkan gabungan spesifikasi dalam standar web, berikut adalah hasil dari W3C:

- 1) Standar web yang paling mendasar adalah HTML, *Cascade Style Sheet* (CSS), dan *Extended Markup Language* (XML).

- 2) Standar HTML yang terakhir adalah *Extended Hypertext markup Language* 1.0 (XHTML 1.0).

### 2.5.2 HTML

HTML atau Hyper Text Markup Language, adalah bahasa yang digunakan untuk mendesain dan memformat halaman web. Kita mungkin sering mendengar tentang bahasa program seperti C, C++, Java, dan Visual basic. Masing-masing bahasa ini terdiri dari perintah sintak dan *programming*. Sintak ini yang sering digunakan programmer untuk memanggil kode. Sangatlah penting mengetahui bagaimana cara menulis kode menggunakan bahasa yang relevan. Lebih dari itu, kita harus konvensional dengan aturan menyangkut bahasa tertentu. Didalam HTML, sintak ini disebut tag. Tag ditulis dengan tanda-kurung bersudut <sintak>. Ada kelompok tag yang sudah dikenal didalam HTML, yang mana digunakan untuk berbagai tujuan. Sebagai contoh, dalam rangka memodifikasi satu baris teks ke dalam bold, kita menerapkan tag bold dengan suatu tag <b>, kemudian tulis beberapa teks atau suatu paragraph yang berisi beberapa teks, dan tutup tag menggunakan tag </b> seperti dalam contoh listing 1. Semua tag didalam HTML harus ditutup menggunakan sintak </sintak>. Tetapi ada beberapa pengecualian pada aturan ini. Kita memakai tag ini sebab HTML bukanlah bahasa yang sensitip seperti C++ dan Java.

### 2.5.3 CSS (Cascading Style Sheet)

CSS memungkinkan web developer untuk memisahkan HTML dari aturan-aturan untuk membentuk tampilan sebuah website. CSS (*Cascading Style Sheet*) digunakan untuk melengkapi file HTML, dan tugas utamanya adalah menetapkan aturan tampilan/style yang akan digunakan pada sebuah website. CSS diperkenalkan untuk pengembangan website pada tahun 1996. Nama CSS didapat dari fakta bahwa setiap deklarasi *style* yang berbeda dapat diletakkan secara berurutan, yang kemudian akan membentuk hubungan parent-child pada setiap *style*, setelah CSS distandarisasikan, *Internet Explorer* dan *Netscape* melepas *browser* terbaru yang telah sesuai atau paling tidak hampir mendekati dengan standar CSS.

CSS adalah sebuah dokumen yang berdiri sendiri dan dapat dimasukkan dalam kode HTML atau sekedar mejadi rujukan oleh HTML dalam pendefinisian *style*. CSS menggunakan kode yang tersusun untuk menetapkan style pda elemen HTML atau dapat juga digunakan membuat *style* baru yang biasa disebut *class*. CSS dapat mengubah besar kecilnya *text*, mengganti warna *background* pada sebuah halaman, atau dapat pula mengubah warna *border* pada tabel, dan masih banyak lagi hal yang dapat dilakukan oleh CSS. Singkatnya, CSS digunakan untuk mengatur susunan tampilan pada halaman HTML. CSS dapat digunakan untuk menggantikan <font>, <b>, <u> dan <u>, dikarenakan hal berikut:sebuah *file* css dapat menjadi rujukan banyak halaman HTML. Hanya dibutuhkan 1 baris kode untuk melakukan hal tersebut. Ini berarti akan meminimalkan file HTML yang akan dibuat. Jika ingin mengubah tampilan website yang telah dibuat, maka yang perlu dilakukan hanya mengganti baris-baris kode pada css nya saja, tanpa perlu mengutak-atik file-file HTML nya, CSS dapat mengatur banyak atribut pada sebuah halaman secara mudah. Misalnya: warna *background*, *border*, *shadow*, yang berbeda pada masing-masing tag yang digunakan.

#### **2.5.4 PHP (PHP Hypertext Preprocessor)**

PHP adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (*Form Interpreted*), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data *form* dari web. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI, kependekan dari *Hypertext Preprocessing*/*Form Interpreter*. Dengan perilsan kode sumber ini menjadi *open source*, maka banyak *programmer* yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan. Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang *interpreter* PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998,

perusahaan tersebut merilis *interpreter* baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi. Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Kelebihan PHP meliputi :

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung PHP dapat ditemukan dimana - mana dari mulai IIS sampai dengan apache, dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (linux, unix, windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

### **2.5.5 Web Service**

*Web service* merupakan aplikasi yang dibuat agar dapat dipanggil atau diakses oleh aplikasi lain melalui internet dengan menggunakan *Extensible Markup Language* (XML) (Lucky, 2008). Selanjutnya dapat diartikan bahwa *web service* adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan

(*service*) yang disediakan oleh suatu sistem yang menyediakan web service. *Web service* menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda *platform*, sistem operasi, maupun bahasa compiler. *Web service* bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam *Web Service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya. Beberapa alasan mengapa digunakannya web service adalah sebagai berikut:

1. *Web service* dapat digunakan untuk mentransformasikan satu atau beberapa *bisnis logic* atau *class* dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. *Web service* memiliki kemudahan dalam proses *deployment*-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. *Web service* cukup di-upload ke *web server* dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. *Web service* berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian *web service* tidak memerlukan konfigurasi khusus di sisi *firewall*.

Web service memiliki tiga entitas dalam arsitekturnya, yaitu:

1. *Service Requester* (peminta layanan)

Peminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.

2. *Service Provider* (penyedia layanan)

Berfungsi untuk menyediakan layanan/*service* dan mengolah sebuah registry agar layanan-layanan tersebut dapat tersedia.

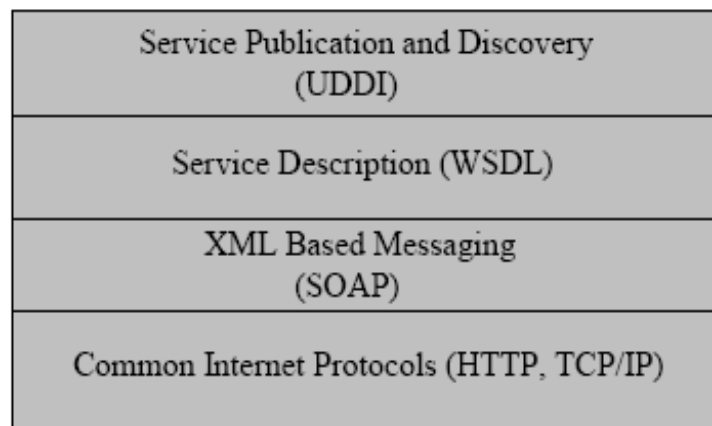
3. *Service Registry* (daftar layanan)

Berfungsi sebagai lokasi central yang mendeskripsikan semua layanan/*service* yang telah di-*register*.

Web service memiliki tiga operasi yang terlibat di dalamnya, yaitu:

1. *Publish/Unpublish* berfungsi menerbitkan/menghapus layanan ke dalam atau dari *registry*.
2. *Find* berfungsi sebagai service requestor mencari dan menemukan layanan yang dibutuhkan.
3. *Bind* berfungsi sebagai service requestor setelah menemukan layanan yang dicarinya, kemudian melakukan *binding* ke *service provider* untuk melakukan interaksi dan mengakses layanan/service yang disediakan oleh *service provider*.

Web Service mencakup empat layer komponen seperti pada gambar berikut:



Gambar 2.10 Layer *Web Service* (Sumber: <http://calonprogrammer.blogspot.com/>)

1. Layer 1: Protokol internet standar seperti HTTP, TCP/IP
2. Layer 2: *Simple Object Access Protocol* (SOAP), merupakan protokol akses objek berbasis XML yang digunakan untuk proses pertukaran data/informasi antar layanan.
3. Layer 3: *Web Service Definition Language* (WSDL), merupakan suatu standar bahasa dalam format XML yang berfungsi untuk mendeskripsikan seluruh layanan yang tersedia.

### 2.5.6 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

### 2.5.7 Google Maps API

API atau *Application Programming Interface* bukan hanya satu set class dan method atau fungsi dan signature yang sederhana. Akan tetapi API, yang bertujuan utama untuk mengatasi “*clueless*” dalam membangun *software* yang berukuran besar, berawal dari sesuatu yang sederhana sampai ke yang kompleks dan merupakan perilaku komponen yang sulit dipahami. Secara sederhana dapat dipahami dengan membayangkan kekacauan yang akan timbul bila mengubah *database* atau skema XML. Perubahan ini dapat dipermudah dengan bantuan API (Tulach, 2008).

Dari beberapa sumber yang didapat, dapat disimpulkan bahwa API adalah sekumpulan perintah, fungsi, *class* dan protokol yang memungkinkan suatu *software* berhubungan dengan *software* lainnya. Tujuan dari API adalah untuk menghilangkan “*clueless*” dari sistem dengan cara membuat blok besar yang terdiri dari *software* di seluruh dunia dan menggunakan kembali perintah, fungsi, *class*, atau *protocol* yang mereka atau API miliki. Dengan cara ini, *programmer* tidak perlu lagi membuang waktu untuk membuat dan menulis infrastruktur sehingga akan menghemat waktu kerja dan lebih efisien.

Seperti yang tercatat oleh Svennerberg (*Beginning Google Maps API 3, p1*), Google Maps API yang paling populer di internet. Pencatatan yang dilakukan pada bulan Mei 2010 ini menyatakan bahwa 43% *mashup* (aplikasi dan situs web yang menggabungkan dua atau lebih sumber data) menggunakan Google Maps API . Beberapa tujuan dari penggunaan Google Maps API adalah untuk melihat lokasi, mencari alamat, mendapatkan petunjuk mengemudi dan lain sebagainya. Hampir semua hal yang berhubungan dengan peta dapat memanfaatkan Google Maps. Google Maps (tanpa API) diperkenalkan pada Februari 2005 dan merupakan revolusi bagaimana peta di dalam web, yaitu dengan membiarkan *user* untuk menarik peta sehingga dapat menavigasinya. Solusi peta ini pada saat itu masih baru dan membutuhkan *server* khusus. Beberapa saat setelahnya, ada yang berhasil melakukan *hacking* Google Maps untuk digunakan di dalam webnya sendiri. Hal ini membuat Google Maps mengambil kesimpulan bahwa mereka membutuhkan API dan pada Juni 2005, Google Maps API dirilis secara publik.

## 2.6 Android

Android adalah sebuah sistem informasi untuk perangkat lunak mobile berbasis linux yang mencakup sistem informasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka (Safaat, 2011). Android memiliki empat karakteristik sebagai berikut :

### 1. Terbuka

Android dibangun untuk benar-benar terbuka sehingga sebuah aplikasi dapat dibangun untuk memanggil salah satu inti ponsel seperti membuat panggilan, mengirim pesan teks, menggunakan kamere dan lain-lain. Android menggunakan sebuah mesin *virtual* yang dirancang khusus untuk mengoptimalkan sumber daya memori dan perangkat keras yang terdapat didalam perangkat. Android merupakan *open source* yang dapat secara bebas diperluas untuk memasukan teknologi baru yang lebih maju pada saat teknologi tersebut muncul. *Platform* ini akan terus berkembang untuk membangun aplikasi *mobile* yang inovatif.



2. Semua aplikasi dibuat sama

Android tidak membedakan aplikasi utama pada telepon dengan aplikasi dari pihak ketiga (*third-party application*). Semua aplikasi dapat dibangun untuk memiliki akses yang sama terhadap kemampuan sebuah telepon dalam menyediakan layanan dan aplikasi yang luas terhadap pengguna.

3. Memecahkan hambatan pada aplikasi

Android memecahkan hambatan untuk membangun sebuah aplikasi yang baru dan inovatif. Misalnya, aplikasi dapat dibangun dan dikembangkan dengan memiliki hubungan langsung dengan informasi yang tersedia pada web seperti lokasi geografis dan lain-lain.

4. Pengembangan aplikasi yang cepat dan mudah

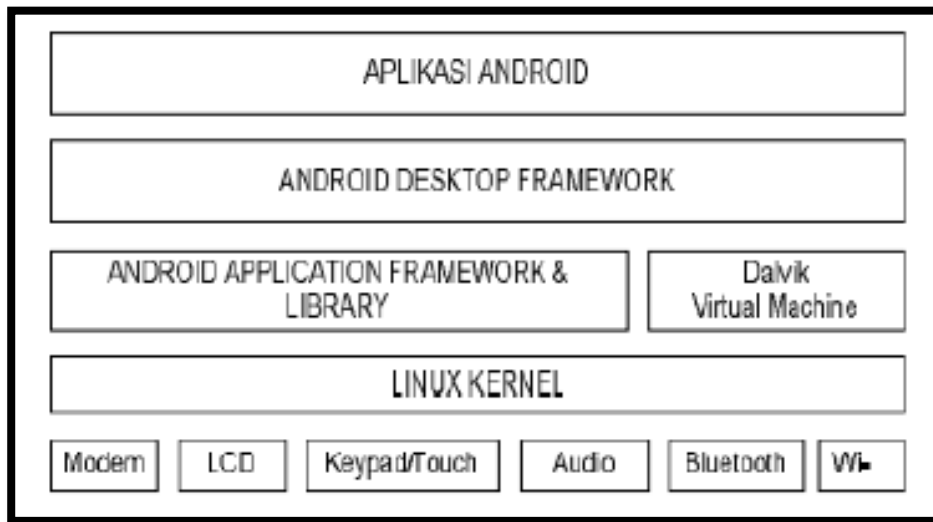
Android menyediakan akses yang sangat luas terhadap pengguna untuk menggunakan *library* yang diperlukan dan *tools* yang sesuai untuk penggunaan aplikasi. Android memiliki sekumpulan *tools* yang dapat digunakan untuk para pengembang dalam meningkatkan kreativitas dan produktivitas aplikasi yang dibuat.

Google *Inc.* secara penuh membangun android dan menjadikannya bersiat terbuka sehingga para pengembang dapat menggunakan android tanpa menggunakan biaya untuk lisensi dari google dan dapat digunakan untuk membangun aplikasi android tanpa ada batasan. *Android Software Development Kit* (SDK) menyediakan alat dan *Application Programming Interface* (API) yang diperlukan untuk memulai aplikasi pada *platform* android menggunakan bahasa java.

Adapun beberapa versi android yang pernah dirilis hingga saat ini adalah sebagai berikut :

1. Android versi 1.1
2. Android versi 1.5 (*Cupcake*)
3. Android versi 1.6 (*Donut*)
4. Android versi 2.0/2.1 (*Eclair*)
5. Android versi 2.2 (*Froyo: Frozen Yoghurt*)
6. Android versi 2.3 (*Gingerbread*)

7. Android versi 3.0/3.1 (*Honeycomb*)
8. Android versi 4.0 (*ICS: Ice Cream Sandwich*)
9. Android versi 4.1 (*Jelly Bean*)
10. Android versi 4.2 (*Jelly Bean*)
11. Android versi 4.2 (*Key Lime Pie*)



Gambar 2.11 Lapisan arsitektur android secara umum

(Suadi-Mazharuddin, 2011)

Adapun beberapa fitur pendukung android sebagai berikut :

#### 1. *Android Software Development Kit* (Android SDK)

Android SDK merupakan *tools* bagi para *programmer* yang ingin mengembangkan aplikasi berbasis google android. Android SDK mencakup seperangkat alat pengembangan yang komprehensif. Android SDK terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, contoh kode dan tutorial. IDE yang didukung secara resmi adalah Eclipse 3.5 atau lebih dengan menggunakan *plugin Android Development Tools* (ADT), dengan ini pengembang dapat menggunakan teks *editor* untuk mengedit *file* Java dan XML serta menggunakan peralatan command line untuk menciptakan, membangun, melakukan *debug* aplikasi Android dan pengendalian perangkat Android.

## 2. *Dalvik Debug Monitor Server (DDMS)*

DDMS merupakan *tools debugging* pada Android yang menyediakan pengambilan gambar layar pada *device*, informasi *thread* dan *heap* pada *device*, *logcat* untuk melihat informasi tentang proses yang sedang berjalan, melihat pesan *error* yang dikirimkan, *warning*, proses, panggilan masuk dan SMS, data lokasi, dan lainnya. DDMS diintegrasikan kedalam Eclipse dan juga terdapat dalam direktori */tools/* pada Android SDK.

## 3. *Android Development Tools (ADT)*

ADT adalah plugin untuk Eclipse yang didesain untuk pengembangan aplikasi Android. ADT memungkinkan Eclipse untuk digunakan dalam membuat aplikasi Android baru, membuat *User Interface*, menambahkan komponen berdasarkan *framework* API Android, *debug* aplikasi, dan pemaketan aplikasi Android.

## 4. *Eclipse Integrated Development Environment (Eclipse IDE)*

IDE adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak. Eclipse adalah sebuah IDE untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*).

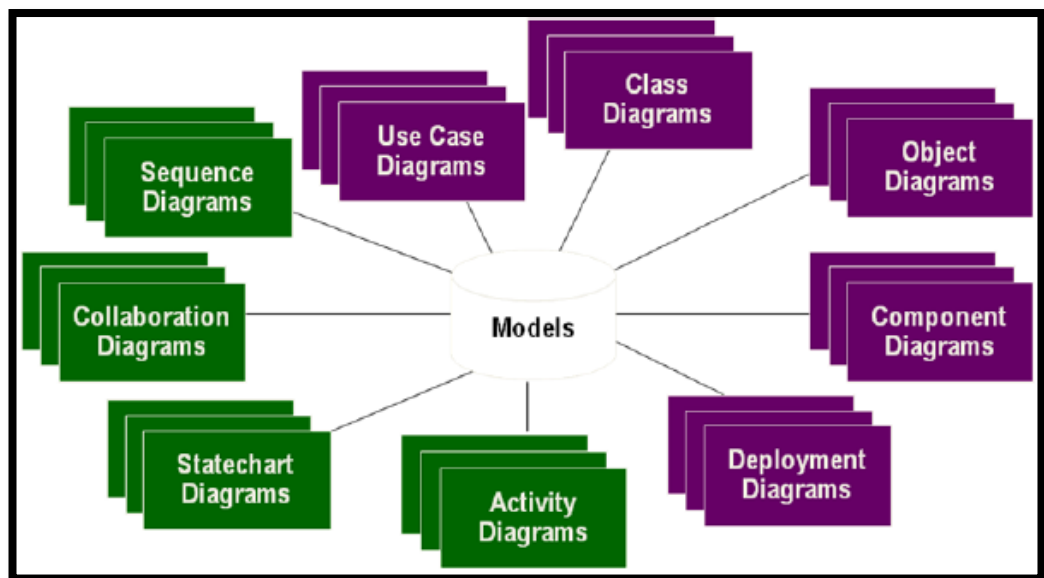
## 2.7 Unified Modeling Language (UML)

*Unified Modelling Language* (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, C# atau VB.NET.

Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB (Visual basic) atau C (Dharwiyanti-Sri, 2003).

UML merupakan bahasa visual untuk menganalisis dan mendesain sebuah sistem berorientasi obyek (Bennet-Steve, 2002). Keunggulan utama yang dimiliki pemodelan ini adalah kemampuannya dalam memodelkan menyerupai kehidupan nyata, sehingga sistem yang dihasilkan mempunyai kelebihan seperti sifat lebih natural, karena umumnya manusia berfikir dalam bentuk objek, pembuatan sistem memakan waktu lebih cepat, dan memudahkan dalam proses pemeliharaan sistem, karena jika ada kesalahan, perbaikan hanya dilakukan pada bagian tersebut, tidak perlu mengurutkan dari awal.

Pada dasarnya UML memuat diagram-diagram pemodelan sistem yang terdiri dari *Use case diagram* (diagram kasus), *Class diagram* (diagram kelas), *Object diagram* (diagram objek), *Statechart diagram* (diagram keadaan), *Activity diagram* (diagram aktivitas), *Sequence diagram* (diagram urutan), *Component diagram* (diagram komponen), *Deployment diagram* (diagram penyebaran), *Collaboration diagram* (diagram kolaborasi) (Nugroho, 2002).

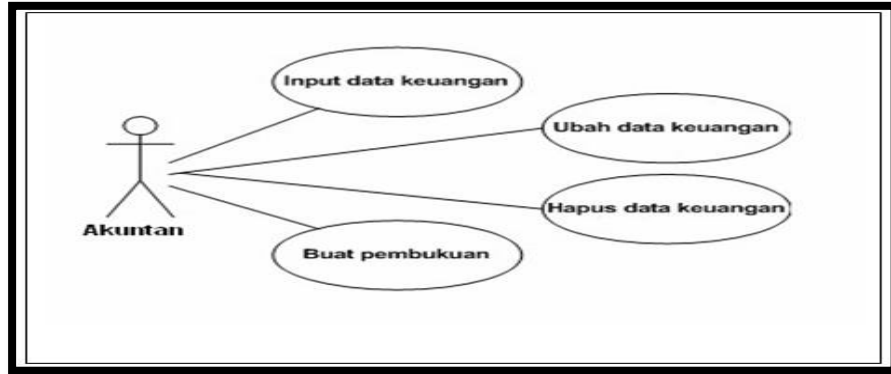


Gambar 2.12. Diagram-diagram pemodelan sistem pada UML  
(Nugroho, 2002).

### 2.7.1 Tipe Diagram UML

#### 1. Use Case Diagram

Menggambarkan sejumlah eksternal aktor dan hubungannya ke use case yang diberikan oleh sistem.



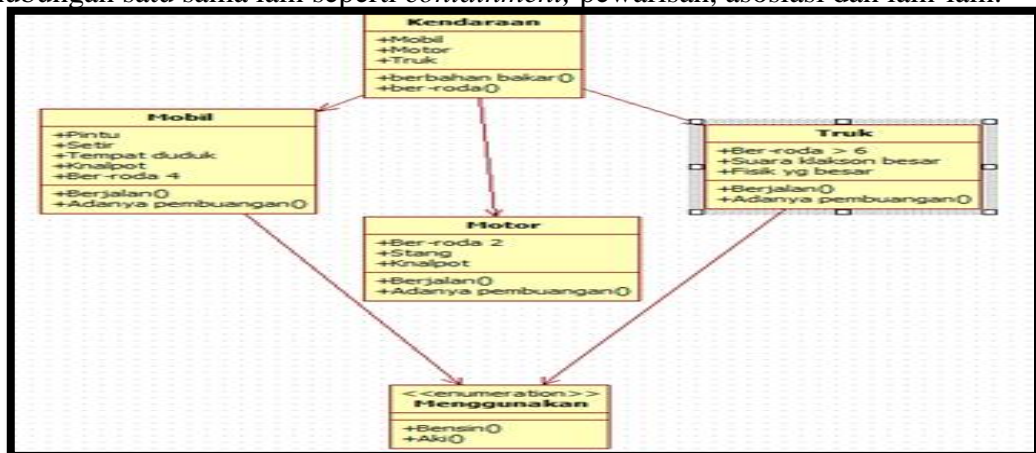
Gambar 2.13. Use Case Diagram

(Sumber: <http://dc301.4shared.com/doc/0bmH6209/preview.html>)

Diagram diatas menerangkan seorang aktor yang bekerja sebagai akuntan pada sebuah perusahaan yang menjalankan sebuah program untuk mengakses database dimana program akan secara otomatis membuat laporan pembukuan dari input akuntan, sehingga manager dapat langsung melihat hasil pembukuan tanpa waktu yang lama. Program tersebut memiliki fitur input data keuangan, ubah data keuangan, hapus data keuangan dan membuat pembukuan.

#### 2. Class Diagram

Menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi dan lain-lain.

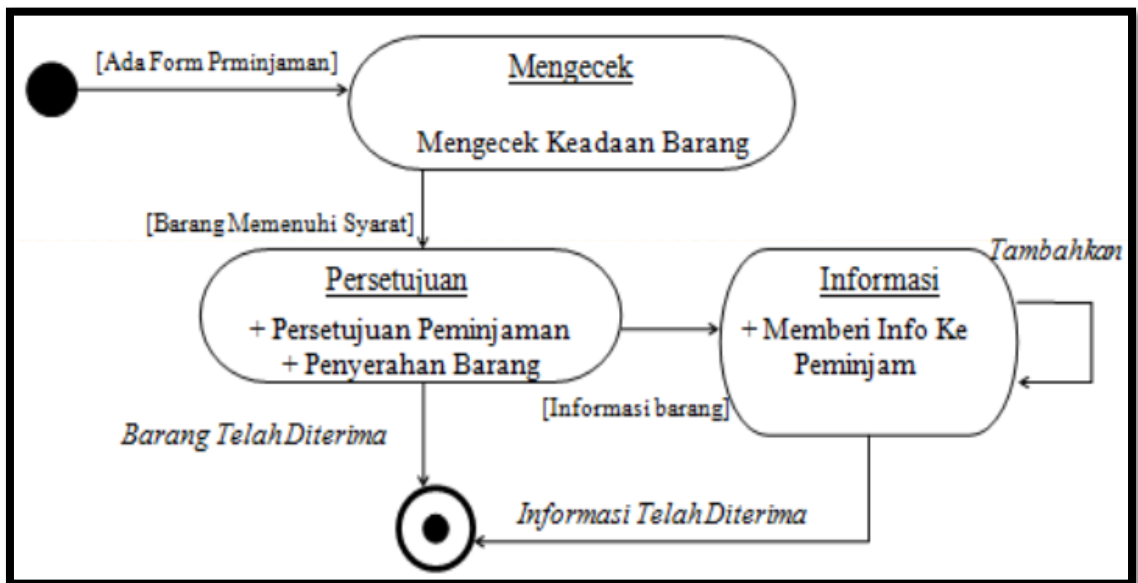


Gambar 2.14. Class Diagram (Sumber: [www.ibm.com](http://www.ibm.com))

Pada *class diagram* diatas menggambarkan ciri-ciri dan bagaimana suatu kendaraan dapat berjalan. Dimulai dari mana saja yang menjadi bagian class kendaraan, lalu merujuk ke ciri-ciri fisik setiap kendaraan tersebut dan menggunakan apa saja agar kendaraan tersebut dapat berjalan.

### 3. Statechart Diagram

Menggambarkan semua state yang dimiliki oleh suatu objek dari suatu *class* dan keadaan yang menyebabkan *state* berubah.

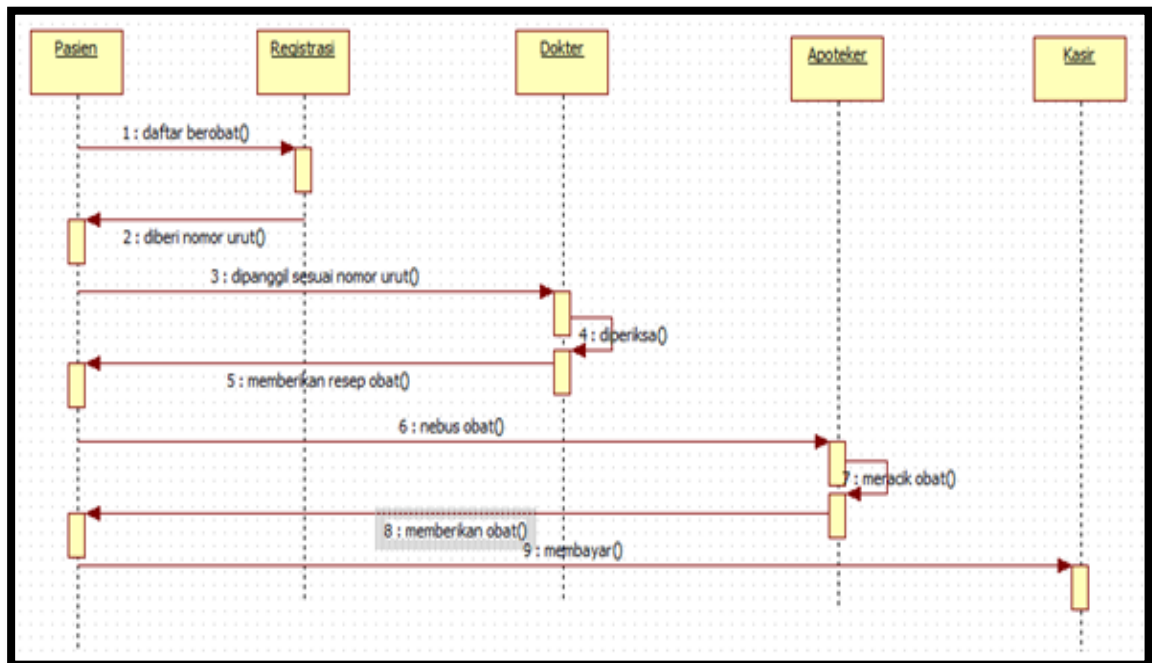


Gambar 2.15. Statechart Diagram (Sumber: <http://rezkiblog.wordpress.com>)

Diagram diatas merupakan kejadian seorang peminjam yang akan meminjam akan mengisi form peminjaman. Sistem akan megecek keadaan barang. Barang tersebut tersedia apa tidak, atau barang tersebut dapat di pinjam atau tidak. Setelah barang tersedia, sistem akan memvalidasi persetujuan peminjaman barang dan menyerahkan barang kepada peminjam. Sistem juga akan mencari informasi tentang barang yang akan dipinjam, maka akan dilakukan permintaan akan informasi barang. Jika informasi yang diterima masih kurang, akan dilakukan permintaan ulang sampai seluruh informasi yang dibutuhkan didapatkan. Saat informasi sudah cukup, informasi tersebut akan diserahkan kepada peminjam barang tersebut.

#### 4. Sequence Diagram

Menggambarkan kolaborasi dinamis antara sejumlah objek dan untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antara objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

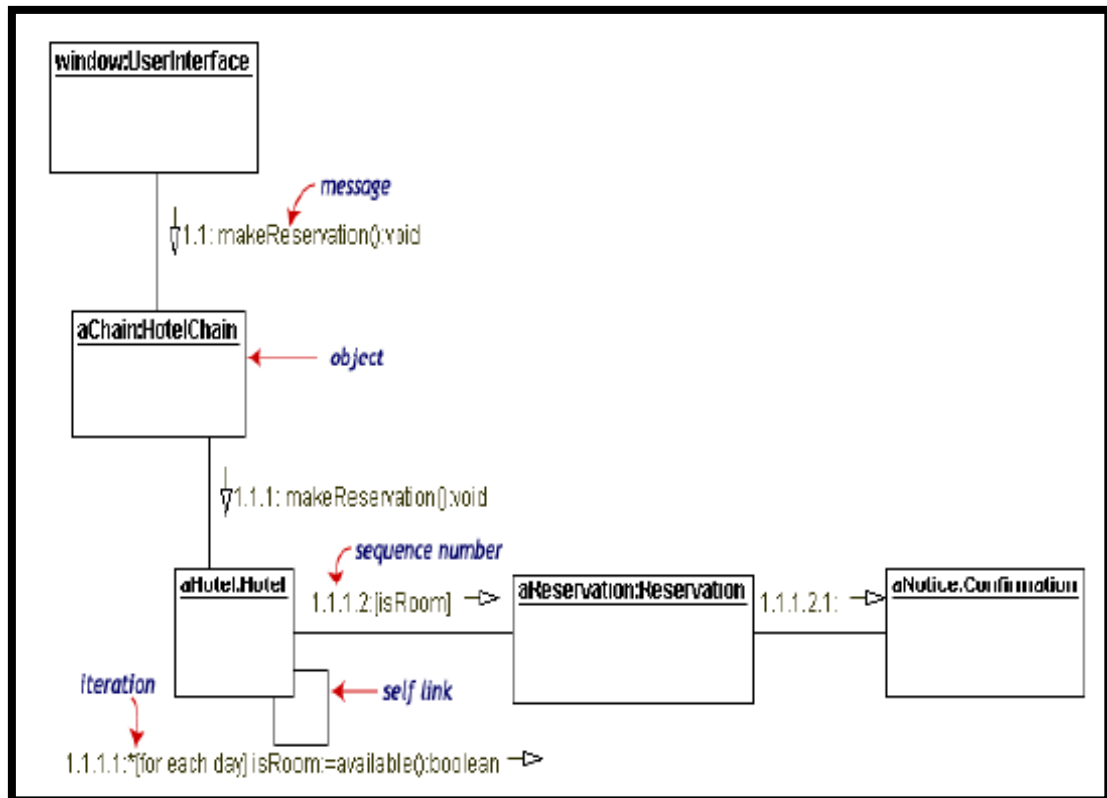


Gambar 2.16. *Sequence Diagram* (Sumber: stackoverflow.com)

Pada *sequence diagram* diatas terdapat 5 aktor dengan tugas mereka masing-masing. Dimulai dari pasien mendaftar ke bagian registrasi, lalu bagian registrasi memberikan nomor urut pasien. Setelah nomor urut didapat, pasien menunggu giliran menemui dokter, dipanggil berdasarkan urutan nomor. Selama didalam ruang dokter, pasien diperiksa penyakitnya. Setelah diketahui, dokter memberikan resep kepada pasien. Pasien menebus obat ke apoteker, bagian apoteker meracik obat sesuai resep yang diberikan dokter. Setelah obat selesai, diberikan kembali ke pasien dan pasien membayar dikasir.

#### 5. Collaboration Diagram

Menggambarkan kolaborasi dinamis seperti *sequence diagram*. Dalam menunjukan pertukaran pesan, *collaboration diagram* menggambarkan objek dan hubungannya



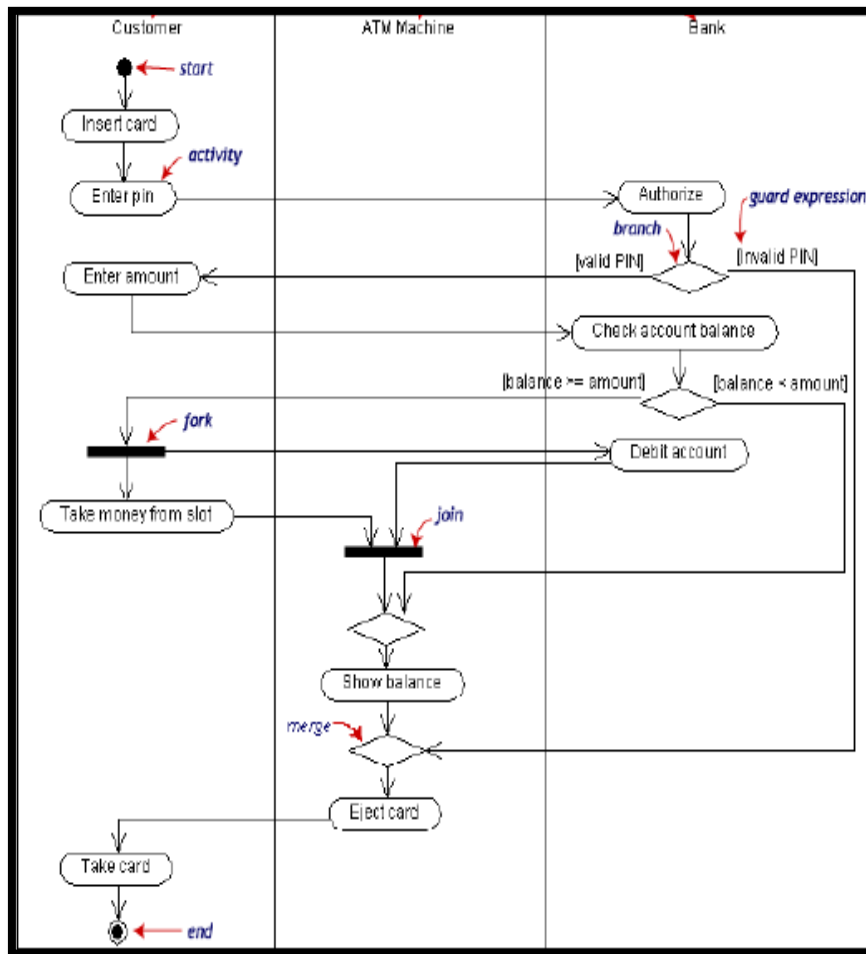
Gambar 2.17. Collaboration Diagram (Sumber: <http://www.visual-paradigm.com>)

Pada contoh collaboration diagram diatas merupakan proses pemesanan kamar hotel. Dimulai dari pemberian pesan dari objek untuk melakukan reservasi. Selanjutnya melakukan pengecekan apakah masih tersedia kamar atau tidak. Apabila masih ada proses dilanjutkan ke reservasi kamar hotel dan dikonfirmasi.

## 6. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *Use Case* atau interaksi.



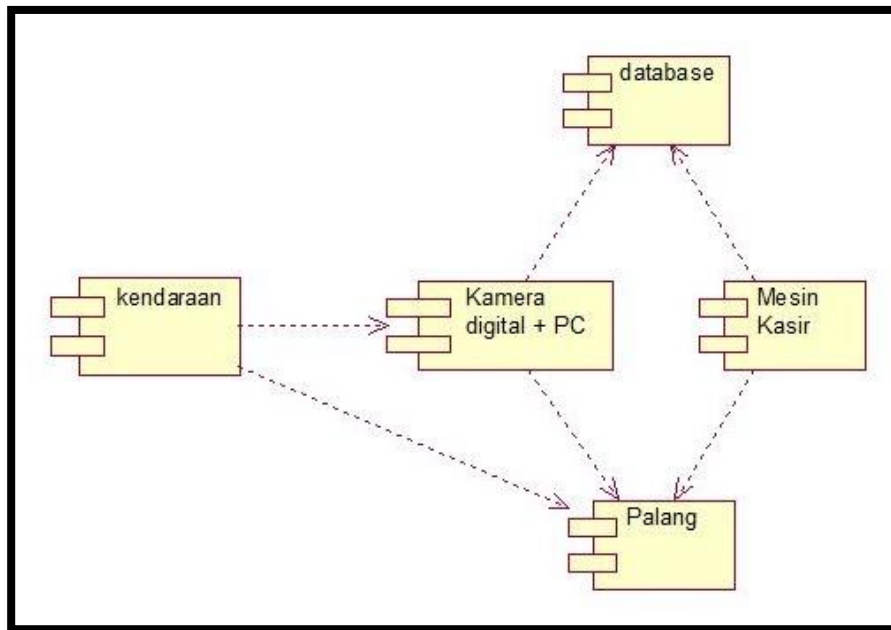


Gambar 2.18 *Activity Diagram* (Sumber: <http://www.visual-paradigm.com>)

Diagram diatas menunjukkan proses pengambilan uang di bank melalui ATM. Ada tiga aktifitas kelas (orang, dan lainnya) yang terkait, yaitu : *Customer*, *ATM*, and *Bank*. Proses berawal dari lingkaran start hitam pada bagian atas dan berakhir di pusat lingkaran stop hitam/putih pada bagian bawah. Aktivitas digambarkan dalam bentuk kotak persegi.

## 7. *Component Diagram*

Menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan diantaranya. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lainnya.

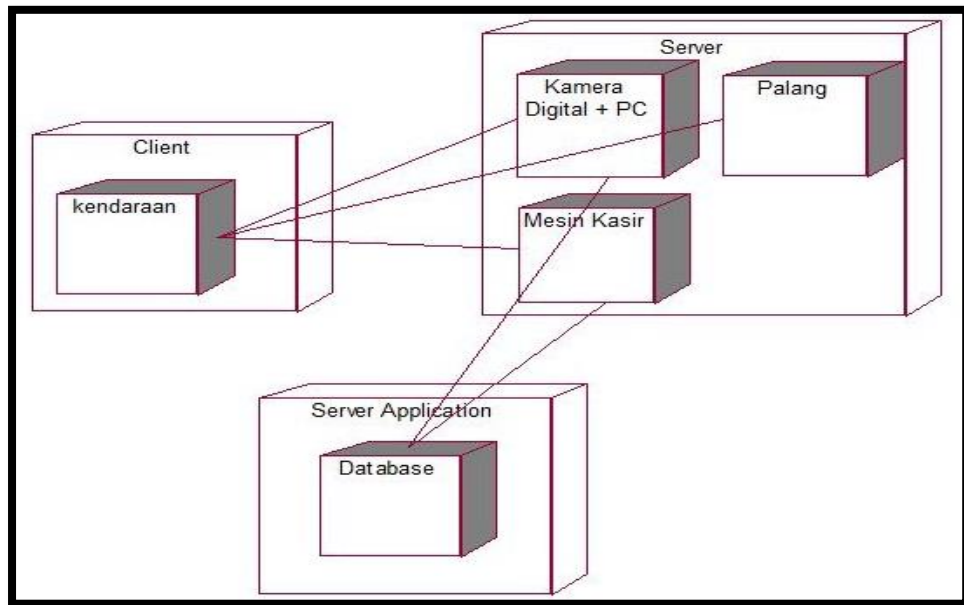


Gambar 2.19. *Component Diagram* (Sumber: <http://www.visual-paradigm.com>)

Diagram diatas menggambarkan alur keluar dan masuk kendaraan pada sistem parkir, berikut penjelasannya : 1. Masuk, Kamera ambil gambar no kendaraan. Lalu penegendara tekan tombol ambil karcis. Lalu mesin mengeluarkan karcis. Lalu pengendara ambil karcis. Kamera digital menyimpan data kendaraan ke database. 2. Setelah pengendara ambil karcis lalu palang terbuka. Pengendara masuk. Lalu parkir kendaraan. 3. Keluar, pengendara kasih karcis kepada mesin kasir / pegawai. Lalu mesin kasir / pegawai mengecek data masuk kendaraan. Setelah mengecek data mesin kasir / pegawai menghitung total biaya parkir. Lalu pengendara bayar total biaya parkir. 4. Setelah pengendara bayar, lalu palang terbuka dan pengendara keluar dari parkir.

#### 8. *Deployment Diagram*

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukan hubungan komputer dengan perangkat satu sama lain dan jenis hubungannya.



Gambar 2.20. Deployment Diagram

(Sumber: [http://www.simventions.com/whitepapers/uml/3000\\_borcon\\_uml.html](http://www.simventions.com/whitepapers/uml/3000_borcon_uml.html))

Diagram *deployment* menggambarkan alur keluar dan masuk kendaraan pada sistem parkir, hampir sama dengan *component diagram* dimulai dari kamera mengambil gambar no kendaraan lalu mesin mengeluarkan karcis, dan pengendara ambil karcis. Kamera digital menyimpan data kendaraan ke database dan palang terbuka. Pengendara keluar dengan memberikan karcis kepada mesin kasir / pegawai, dilakukan pengecekan data masuk kendaraan. Lalu pengendara bayar total biaya parker lalu palang terbuka dan pengendara keluar dari parkir.

## 2.8 Bagan Alir (*FlowChart*)

Bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika (Jogiyanto, 2005). Selanjutnya, Flowcart dapat diartikan sebagai bagan yang mempunyai arus, menggambarkan langkah-langkah penyelesaian suatu masalah. Flowcart merupakan cara penyajian dari suatu algoritma. Pedoman dalam menggambar suatu bagan alir, analisis sistem atau pemrograman sebagai berikut :

1. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan didalam bagan alir harus ditunjukkan dengan jelas.

3. Harus ditunjukkan darimana kegiatan akan dimulai dan berakhir.
4. Masing-masing kegiatan didalam bagan alir sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan.
5. Masing-masing kegiatan didalam bagan alir harus didalam urutan.
6. Kegiatan yang terpotong dan akan disambung ketempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
7. Gunakanlah simbol-simbol bagan alir yang standar.

Menurut Jogiyanto (2005) ada 5 macam bagan alir, diantaranya:

#### 1. Bagan alir sistem (*system flowchart*)

Bagan alir sistem (*system flowchart*) merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan menjelaskan urutan-urutan dari prosedur-prosedur yang ada dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan sistem.

Simbol	Nama	Keterangan	Simbol	Nama	Keterangan
	Simbol Dokumen	Menunjukkan dokumen input dan output baik untuk proses manual, mekanik atau komputer		Simbol drum magnetik	Menunjukkan input/output menggunakan drum magnetik
	Simbol Kegiatan Manual	Menunjukkan pekerjaan manual		Simbol Pita Kertas berlubang	Menunjukkan input/output menggunakan Pita Kertas berlubang
	Simbol Simpanan Offline	File non-komputer yang diarsip		Simbol Keyboard	Menunjukkan input yang menggunakan on-line keyboard
	Simbol Kartu Plong	Menunjukkan input/output yang menggunakan kartu plong ( <i>punched card</i> )		Simbol display	Menunjukkan output
	Simbol Proses	Menunjukkan kegiatan proses dari operasi program komputer		Simbol pita kontrol	Menunjukkan penggunaan pita kontrol ( <i>control tape</i> ) dalam batch control untuk pencocokan di proses batch proses
	Simbol operasi luar	Menunjukkan kegiatan proses di luar proses operasi komputer		Simbol hubungan komunikasi	Menunjukkan proses transmisi data melalui channel komunikasi
	Simbol pengurutan offline	Menunjukkan proses pengurutan data di luar proses komputer		Simbol garis alir	Menunjukkan arus dari proses
	Simbol Pita Magnetik	Menunjukkan input/output menggunakan pita magnetik		Simbol penjelasan	Menunjukkan penjelasan dari suatu proses
	Simbol Hardisk	Menunjukkan input/output menggunakan hard disk		Simbol penghubung	Menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain
	Simbol diskette	Menunjukkan input/output menggunakan diskette			

Gambar 2.21 Notasi Bagan Alir Sistem (Sumber: Jogiyanto, 2005)

b. Bagan alir dokumen (*document flowchart*)









Bagan alir dokumen (*document flowchart*) atau disebut bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Bagan alir dokumen ini menggunakan simbol-simbol yang sama dengan yang digunakan di dalam bagan alir sistem.

c. Bagan alir skematik (*schematic flowchart*)

Bagan alir skematik (*schematic flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Perbedaannya adalah bagan alir skematik menggunakan simbol-simbol bagan alir sistem, juga menggunakan gambar-gambar komputer dan peralatan lainnya yang digunakan. Maksud penggunaan gambar-gambar ini adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol bagan alir.

d. Bagan alir program (*program flowchart*)






Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program.

Simbol	Nama Simbol	Keterangan	Simbol	Nama Simbol	Keterangan
	Simbol input/output	Simbol input/output ( <i>input/output symbol</i> ) digunakan untuk mewakili data input/output		Simbol persiapan	Simbol persiapan ( <i>preparation symbol</i> ) digunakan untuk memberi nilai awal suatu besaran
	Simbol Proses	Simbol proses digunakan untuk mewakili suatu proses		Simbol titik terminal	Simbol titik terminal ( <i>terminal point symbol</i> ) digunakan untuk menunjukkan awal dan akhir dari suatu proses
	Simbol garis alir	Simbol garis alir ( <i>flow lines symbol</i> ) digunakan untuk menunjukkan arus dari proses			
	Simbol penhubung	Simbol penghubung ( <i>connector symbol</i> ) digunakan untuk menunjukkan sambungan dari bagan alir yang terputus di halaman yang sama atau di halaman lainnya			
	Simbol keputusan	Simbol keputusan ( <i>decision symbol</i> ) digunakan untuk suatu penyelesaian kondisi di dalam program			
	Simbol proses terdefinisi	Simbol proses terdefinisi ( <i>predefined process symbol</i> ) digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain			

Gambar 2.22 Notasi Bagan Alir Program (Sumber: Jogiyanto, 2005)

e. Bagan alir proses (*process flowchart*)

Bagan alir proses (*process flowchart*) merupakan bagan alir yang banyak digunakan teknik industri. Bagan alir juga berguna bagi analisis sistem untuk menggambarkan proses dalam suatu prosedur. Bagan alir proses menggunakan lima buah simbol tersendiri.

Simbol	Keterangan
	Menunjukkan suatu operasi ( <i>operation</i> )
	Menunjukkan suatu pemindahan ( <i>movement</i> )
	Menunjukkan suatu simpanan ( <i>storage</i> )
	Menunjukkan suatu inspeksi ( <i>inspection</i> )
	Menunjukkan suatu penundaan ( <i>delay</i> )

Gambar 2.23 Notasi Bagan Alir Proses (Sumber: Jogiyanto, 2005)

## 2.9 Hasil Penelitian Terkait

Merujuk terhadap judul penelitian yang diangkat terdapat beberapa teori serta pengetahuan dasar mengenai penelitian terkait dengan konsep maupun tujuan yang sejalan. Adapun beberapa penelitian yang terkait meliputi :

1. Penelitian yang dilakukan oleh Sutra Handoko Jurusan Teknik Informatika, Fakultas Teknik, Universitas Bengkulu tahun 2010 dengan judul “Manajemen Basis Data Spasial Kawasan Hutan Beserta Ekosistemnya Wilayah Provinsi Bengkulu Dengan Menggunakan SIG Berbasis Web”. Penelitian ini berfokus pada Web manajemen basis data spasial kawasan hutan mempresentasikan kawasan hutan Provinsi Bengkulu beserta persebaran ekosistem dan aksesibilitas jalannya. Sistem ini menyajikan data spasial berupa kawasan hutan, persebaran ekosistem dan aksesibilitas jalan dan serta data non spasial yang terkait.
2. Penelitian yang dilakukan oleh Budi Santosa, Herry Sofyan dan Wahyu Ari Widyastuti, Jurusan Teknik Informatika, UPN Veteran Yogyakarta tahun 2008 dengan judul “Sistem Informasi Geografis Penyebaran Penduduk Berdasarkan

Tingkat Usia di Kabupaten Selman Berbasis Web”. Sistem ini akan memudahkan instansi terkait untuk mengetahui potensi penduduk (usia produktif atau non-produktif) dari suatu daerah, juga menyediakan informasi bagi para pegambil kebijakan tingkat Propinsi dan Kabupaten untuk menghasilkan perencanaan yang lebih baik dalam pemanfaatan potensi suatu daerah dalam kaitannya dengan pengembangan wilayah yaitu dengan memperhatikan faktor-faktor seperti iklim, kesuburan tanah, tata air, keadaan ekonomi serta kultur penduduk (mata pecahaiannya).

3. Penelitian yang dilakukan oleh Edi Irwansyah, Sena Adhinugraha dan Tri Datar Wijaya, Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Bina Nusantara Jakarta tahun 2011 dengan judul “Pengembangan Sistem Informasi Geografis (SIG) Pada Platform Google Untuk Penanggulangan Kebakaran di Jakarta Selatan”. Penelitian yang dilakukan bertujuan untuk mengembangkan Sistem Informasi Geografis pada Platform Google mengenai letak posisi daerah rawan kebakaran di Jakarta Selatan beserta titik-titik penanggulangan dan juga sebagai alat analisis untuk penambahan dan pengembangan komponen pendukung dalam penanganan dan penanggulangan bencana kebakaran.
4. Penelitian yang dilakukan oleh Erin Utami, Arna Fariza dan Rengga Asmara, Politeknik Elektronika Negeri Surabaya, Institut Teknologi Sepuluh Nopember tahun 2011 dengan judul “ Aplikasi Sistem Informasi Geografis Tentang Kos-Kosan Menggunakan Google Maps API 3 (Studi Kasus : Surabaya bagian timur). Penelitian ini berfokus pada sistem informasi geografis letak kos-kosan khususnya daerah surabaya timur yang mana didaerah tersebut terdapat banyak sekali universitas yang sebagian besar mahasiswanya berasal dari luar kota surabaya sehingga nantinya sistem informasi geografis ini dapat membantu mahasiswa dan masyarakat yang membutuhkan kos-kosan. Sistem yang saya bangun ini dapat memvisualisasikan jalur dari tempat asal ke kos-kosan dan dilengkapi navigasi arah yang berfungsi mencari jalur terpendek yang terdapat perhitungan jarak (kilometer) dari tempat asal ke kos-kosan dan mencari lokasi kos-kosan menggunakan Google Maps API 3.

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Jenis Penelitian**

Jenis penelitian yang dilakukan adalah penelitian deskriptif dengan menggunakan pendekatan kualitatif melalui survei, observasi, wawancara ahli dan studi dokumentasi. Penelitian deskriptif ini bertujuan untuk memberikan gambaran yang sistematis, cermat, dan akurat mengenai kegiatan konservasi di Provinsi Bengkulu. Penelitian deskriptif juga ditujukan untuk mendeskripsikan fenomena-fenomena yang ada, baik fenomena alamiah maupun fenomena buatan manusia. Fenomena itu bisa berupa bentuk, aktivitas, karakteristik, perubahan, hubungan, kesamaan, dan perbedaan antara fenomena yang satu dengan fenomena lainnya (Sukmadinata, 2006). Penelitian deskriptif merupakan penelitian yang berusaha mendeskripsikan dan menginterpretasikan sesuatu, misalnya kondisi atau hubungan yang ada, pendapat yang berkembang, proses yang sedang berlangsung, akibat atau efek yang terjadi, atau tentang kecenderungan yang tengah berlangsung. Fenomena disajikan secara apa adanya hasil penelitiannya diuraikan secara jelas dan gamblang tanpa manipulasi oleh karena itu penelitian ini tidak adanya suatu hipotesis tetapi adalah pertanyaan penelitian. Analisis deskriptif dapat menggunakan analisis distribusi frekuensi yaitu menyimpulkan berdasarkan hasil rata-rata. Hasil penelitian deskriptif sering digunakan, atau dilanjutkan dengan melakukan penelitian analitik.

#### **3.2 Sumber Data**

Sumber data penelitian ini terdiri atas sumber data sekunder dan data primer. Adapun penjelasan mengenai sumber data penelitian meliputi:

1. Data Primer

Data primer merupakan data yang diperoleh langsung di lapangan oleh peneliti sebagai obyek penulisan. Data primer merupakan sumber data yang diperoleh langsung dari sumber asli (tidak melalui media perantara) (Umar, 2003). Data primer dapat berupa opini subjek (orang) secara individual atau kelompok,



hasil observasi terhadap suatu benda (fisik), kejadian atau kegiatan, dan hasil pengujian. Data primer dalam penelitian ini diperoleh dari observasi lapangan terkait kegiatan konservasi. Selain itu, data primer ditunjang oleh hasil wawancara informan maupun pihak yang dianggap mengetahui latar belakang kasus seperti informan pada Balai Konservasi dan Sumber Daya Alam (BKSDA) Provinsi Bengkulu dan Informan Pusat Informasi Konservasi Alam Bogor. Data Primer ini berupa observasi lapangan secara langsung mengenai koordinat kegiatan konservasi dan dalam bentuk foto fisik.

## **2. Data Sekunder**

Data sekunder adalah data yang tidak langsung memberikan data kepada peneliti, misalnya penelitian harus melalui orang lain atau mencari melalui dokumen (Sugiyono, 2005). Data ini diperoleh dengan menggunakan studi literatur yang dilakukan terhadap banyak buku dan diperoleh berdasarkan catatan-catatan yang berhubungan dengan penelitian, selain itu peneliti mempergunakan data yang diperoleh dari internet. Dalam hal ini, data sekunder bersumber dari BKSDA Provinsi Bengkulu dan Dinas Kehutanan Provinsi Bengkulu. Data tersebut berupa data aktivitas konservasi yang dilakukan oleh Polhut dalam beberapa waktu terakhir dimana berisi temuan fauna, flora, koordinat kegiatan, foto lapangan dan kegiatan konservasi lainnya. Data sekunder dapat digunakan sebagai pelengkap untuk mendukung informasi dari data primer yang telah dikumpulkan. Untuk data lengkap mengenai kegiatan konservasi ada pada lampiran penulisan.

## **3.3 Metode Penelitian**

### **3.3.1 Metode Pengumpulan Data**

Dalam pelaksanaan penelitian, metode pengumpulan data yang digunakan adalah Wawancara, Observasi dan Studi Pustaka.

#### **1. Wawancara**

Pengumpulan data dilakukan dengan wawancara melalui dua pendekatan yaitu emik dan etik. Emik dimaksudkan untuk mengumpulkan seluruh informasi kegiatan konservasi melalui sudut pandang informan. Sedangkan etik dimaksudkan untuk melakukan analisis berdasarkan disiplin keilmuan. Dalam penelitian ini

wawancara dilakukan dengan teknik terstruktur dan bebas. Dalam wawancara terstruktur sudah ada konsep-konsep yang akan digali dan perjanjian langkah-langkah wawancara selanjutnya dengan informan. Wawancara bebas merupakan wawancara tidak terstruktur sesuai situasi.

## 2. Observasi

Observasi lapangan dan pengambilan *sampling* data dilakukan berdasarkan keterangan yang diperoleh dari informan. Observasi lapangan meliputi pengamatan di lokasi, pengamatan di lokasi meliputi pengambilan data GPS seperti koordinat lintang dan bujur yang dilakukan oleh Polhut maupun penulis sendiri.

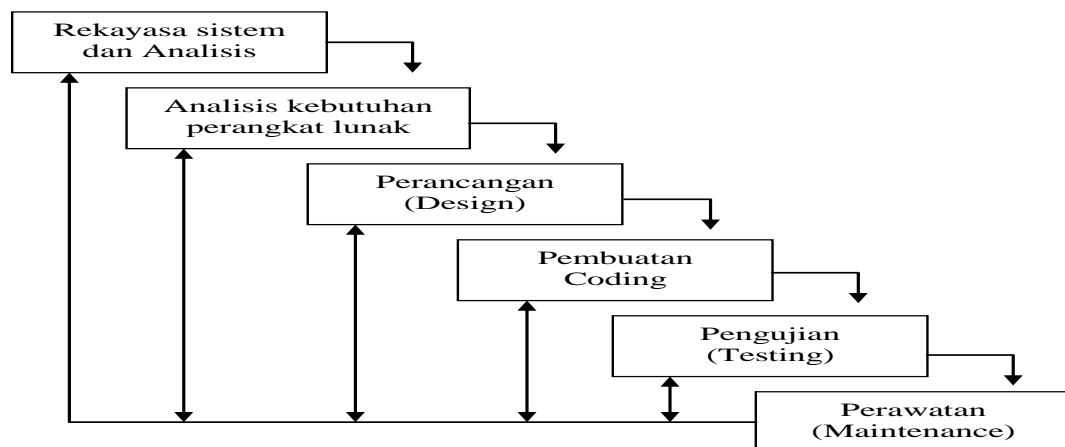
## 3. Studi Pustaka

Studi Pustaka merupakan metode pengumpulan data yang berasal dari buku-buku literatur. Metode Studi pustaka merupakan metode pengumpulan data yang memanfaatkan buku atau literatur sebagai bahan referensi untuk memperoleh kesimpulan-kesimpulan atau pendapat para ahli (Gorys Keraf, 1997). Metode ini dapat dijadikan media akan kurangnya sumberdaya pengetahuan penulis dalam pencarian referensi terkait pengumpulan data maupun perancangan aplikasi yang akan dibangun. Dalam hal ini literatur terkait sebaran flora, fauna dan aktivitas konservasi menjadi kata kunci studi pustaka.

### 3.3.2 Model Pengembangan Sistem

Untuk mencapai tujuan penelitian yang dapat mengakomodir kebutuhan sistem pengembangan basis data dalam inventarisasi penelitian yang telah dilaksanakan, maka setelah itu diperlukan langkah-langkah pengembangan sistem. Langkah-langkah yang dapat dilakukan antara lain: perencanaan sistem, analisis sistem, perancangan dan implementasi sistem. Sistem yang dikembangkan dalam penelitian ini menggunakan model pengembangan sistem SDLC (*System Development Life Cycle*) atau Daur Hidup Pengembangan Sistem. Daur hidup pengembangan sistem adalah pendekatan melalui beberapa tahap untuk menganalisis dan merancang sistem yang dimana sistem tersebut telah dikembangkan dengan sangat baik melalui penggunaan siklus kegiatan penganalisis dan pemakai secara spesifik (Jogiyanto, 2005).

Model yang digunakan adalah model *Waterfall*, atau sering disebut juga model Sekuensial Linear. Pada model ini, sistem akan dibuat secara sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode dan pengujian. Beberapa kelebihan model *waterfall* adalah menyediakan dokumentasi untuk tiap tahapan, tahapan sistematis, dan melakukan pemeriksaan keluaran setiap tahapan. Model *waterfall* dapat dilihat pada Gambar 3.1 berikut:



Gambar 3.1 Model Waterfall (Sommerville, 2003)

#### 1. Analisis dan Perencanaan

Langkah ini bertujuan untuk menganalisa semua komponen-komponen yang dibutuhkan oleh sistem yang akan dibangun, termasuk didalamnya analisa kebutuhan data. Data-data yang dibutuhkan diperoleh melalui tahapan sebagai berikut: Observasi, studi pustaka, dan wawancara. Data dimaksud meliputi data kegiatan konservasi: temuan, pelanggaran, kejadian, tanda batas, penyuluhan, sosialisasi, koordinasi, dan pemasangan papan peringatan di *resort* yang ada. Adapun *resort* tersebut terdiri dari *resort* Kota, Bandara, Sukaraja, Tais, Manna, Kaur, Kepahiang, Bukit Kaba, Lebong, Seblat, Mukomuko dan Enggano.

#### 2. Desain

Pada tahapan ini, tim pengembangan sistem mendefinisikan proses-proses dan kebutuhan-kebutuhan sistem yang berkaitan dengan pengembangan aplikasi (*software development process*). Dalam fase ini ditentukan pemilihan teknologi yang akan diterapkan baik berupa *website integrated technology*, rancangan

*database*, maupun beragam aspek lainnya yang berkaitan dengan kegiatan analisis dan perancangan ini.

- 1) Membuat desain awal dari sistem
- 2) Membuat desain *database* : perancangan *database* yang dibutuhkan oleh sistem yang akan dibuat menggunakan *SQLite* dan *MySQL*.
3. Implementasi

- 1) Kode

Pada tahapan ini hasil dari fase-fase sebelumnya dituangkan kedalam penulisan kode-kode dengan menggunakan bahasa pemrograman komputer yang telah ditentukan dalam tahap sebelumnya. Untuk melakukan pemrograman ini dibutuhkan perangkat-perangkat pemrograman seperti *Eclipse*, *Notepad++*, *Browser*, *MySQL* dan *SQLite* serta aneka perangkat lunak berkaitan lainnya sesuai dengan kebutuhan pemrograman bersangkutan.

- 2) Pengujian

Setelah proses penulisan kode pemrograman langkah berikutnya berupa proses pengujian terhadap hasil pemrograman tersebut . Pengujian mencakup beragam aspek yang berkaitan dengan *System and Performance* dari fase *Code Generation*. Pengujian-pengujian tersebut berupa Pengujian Database, Pengujian Validitas Data, Pengujian Logic Aplikasi, Pengujian Antar Muka Aplikasi (*General User Interface/GUI*), Pengujian User Administration. Hasil pengujian ini merupakan Umpan balik perbaikan *System and Performance* yang akan digunakan dalam proses perbaikan sistem hingga mencapai hasil yang diharapkan dan telah ditentukan sebelumnya. Adapun langkahnya adalah sebagai berikut:

- a. Pengujian unit: menguji kinerja dari masing-masing unit yang telah dirancang dan dilakukan peng-kodingan
- b. Pengujian sistem : bagian-bagian dihubungkan bersama-sama dengan menggunakan data uji untuk mengetahui apakah data / sistem tersebut dapat berjalan sesuai dengan kebutuhan analisa, pengkodingan.
- c. Pengujian user : pengujian ini dilakukan terhadap pengguna dari sistem yang telah dibangun, meliputi : pelatihan bagaimana menggunakan sistem, dan menelusuri kekurangan yang terjadi dalam sistem yang digunakan.

### 3) *Maintenance* / pemeliharaan

Tahapan ini melengkapi keseluruhan dari siklus SDLC (*System Development Life Cycle*), dalam pemeliharaan sistem ini dilakukan penyesuaian dan peningkatan sistem yang dilakukan secara audit dan periodik dengan membuat perubahan-perubahan berdasarkan dengan kondisi terbaru.

## 3.4 Model Pengujian Sistem (*Black Box Testing*)

Pada siklus hidup pengembangan perangkat lunak (*System Development Life Cycle: SDLC*) salah satu proses yang harus dilakukan adalah proses pengujian (*testing*). Pengujian perangkat lunak adalah suatu teknik yang digunakan menguji apakah sebuah perangkat lunak yang dihasilkan telah memenuhi kebutuhan proses bisnis pengguna atau masih belum. Menurut Pressman (2005), *testing* adalah proses eksekusi suatu program untuk menemukan kesalahan sebelum digunakan oleh pengguna akhir (*end-user*).

Salah satu metode pengujian perangkat lunak adalah *Black-Box Testing*. *Black-box Testing* merupakan sebuah metode yang digunakan untuk menemukan kesalahan dan mendemonstrasikan fungsional aplikasi saat dioperasikan, apakah input diterima dengan benar dan output yang dihasilkan telah sesuai dengan yang diharapkan. Fokus dari pengujian menggunakan metode *Black-Box* adalah pada pengujian fungsionalitas dan output dihasilkan aplikasi. Pengujian *black-box* didesain untuk mengungkap kesalahan pada persyaratan fungsional dengan mengabaikan mekanisme internal atau komponen dari suatu program.

Pengujian perangkat lunak mempunyai beberapa level, untuk pengujian menggunakan metode *Black Box*, terdapat enam level yaitu *Integration*, *Functional*, *System*, *Acceptance*, *Beta*, dan *Regression* (Williams, 2006). Salah satu dari pengujian *Black-Box* yang dapat dilakukan oleh seorang penguji independen adalah *Functional testing*. Basis uji dari *functional testing* ini adalah pada spesifikasi dari komponen perangkat lunak yang akan diuji. *Functional testing* memastikan bahwa semua kebutuhan-kebutuhan telah dipenuhi dalam sistem aplikasi. Dengan demikian fungsinya adalah tugas-tugas yang didesain untuk

dilaksanakan sistem. *Functional testing* berkonsentrasi pada hasil dari proses, bukan bagaimana prosesnya terjadi.

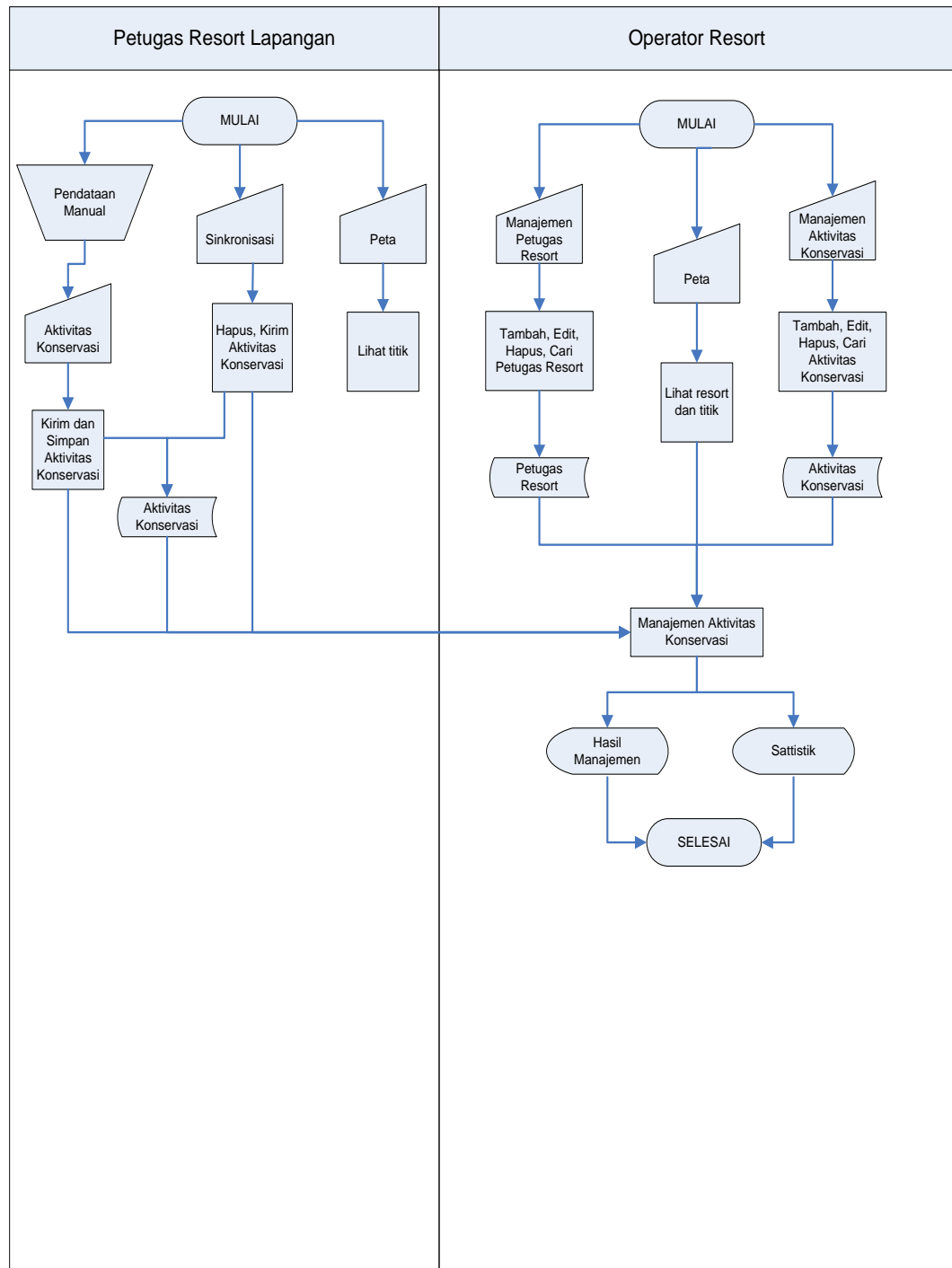
Metode ujicoba *blackbox* memfokuskan pada keperluan fungsional dari software. Karna itu ujicoba *blackbox* memungkinkan pengembang software untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba *blackbox* bukan merupakan alternatif dari ujicoba analisis *internal* kode (*whitebox*), tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *whitebox*. Ujicoba *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan performa.
5. kesalahan inisialisasi dan terminasi

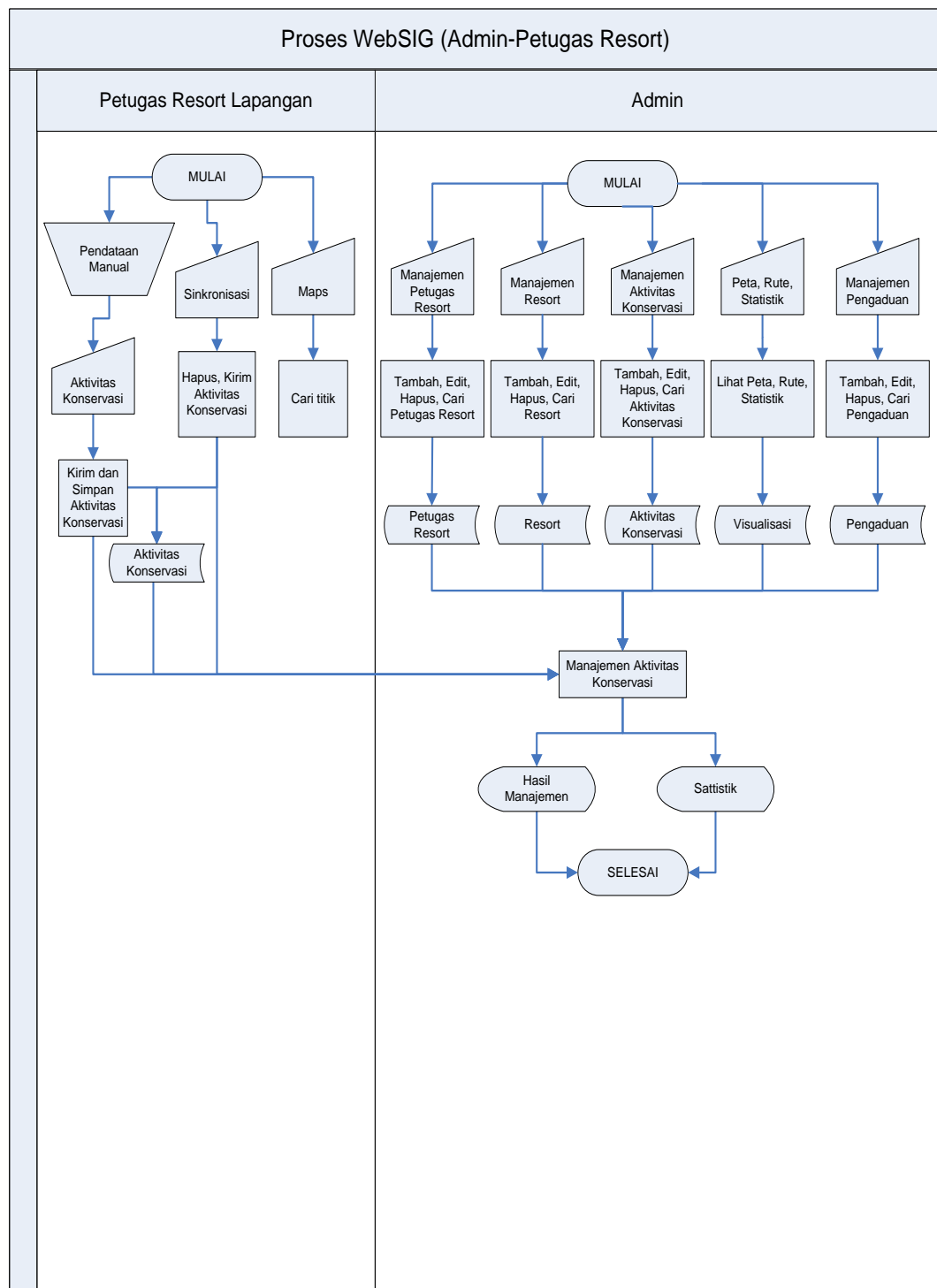
### **3.5 Bagan Alir Sistem (*Flowchart*)**

Adapun bagan alir (*flowchart*) dari sistem yang dibuat terbagi menjadi 4 bagian yang saling berhubungan berdasarkan aktor yang mempengaruhi sistem. Berikut bagan alir dari rancangan sistem :

### Proses WebSIG (Operator-Petugas Resort Lapangan)

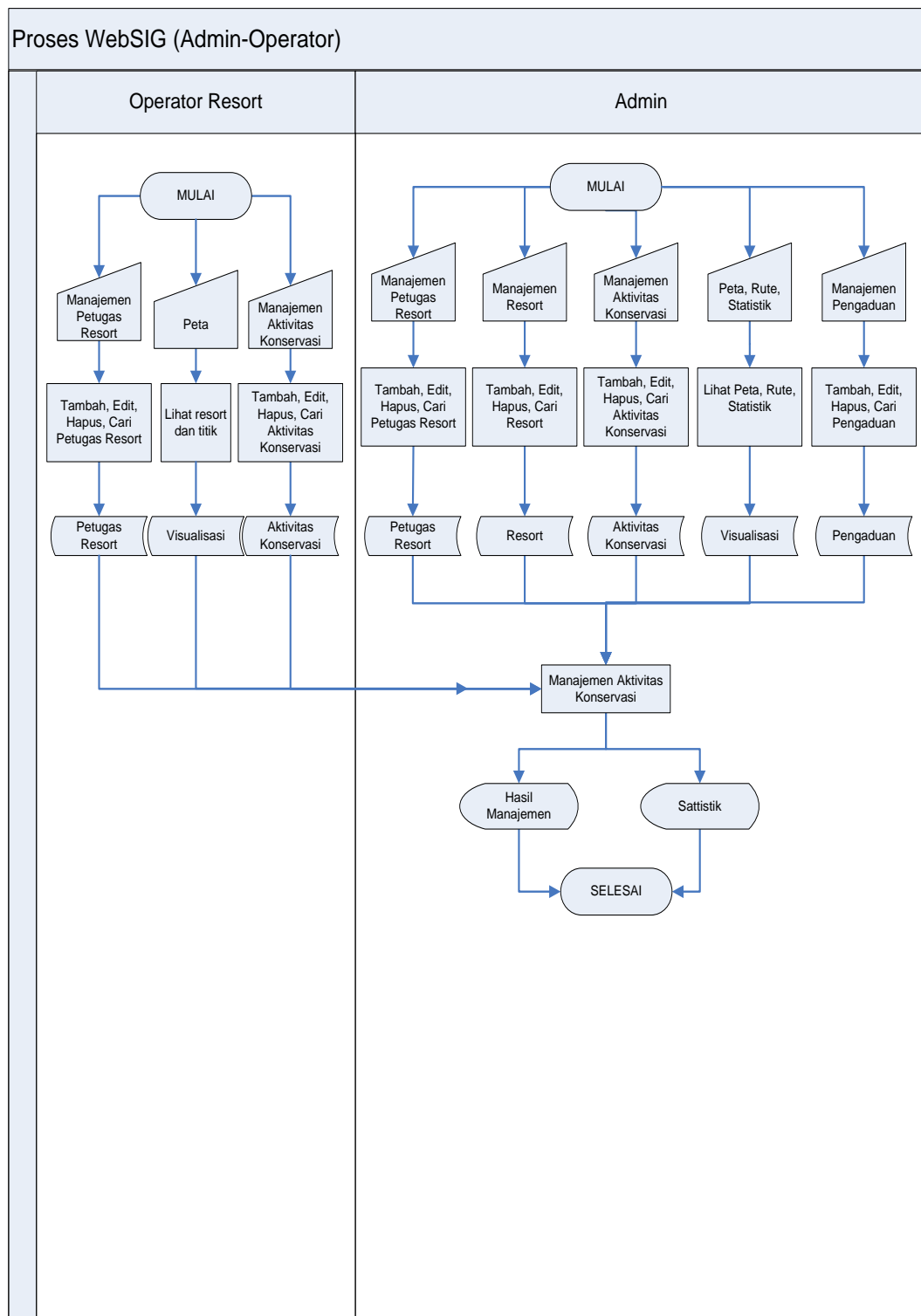


Gambar 3.2 Flowchart sistem operator resort dan petugas lapangan

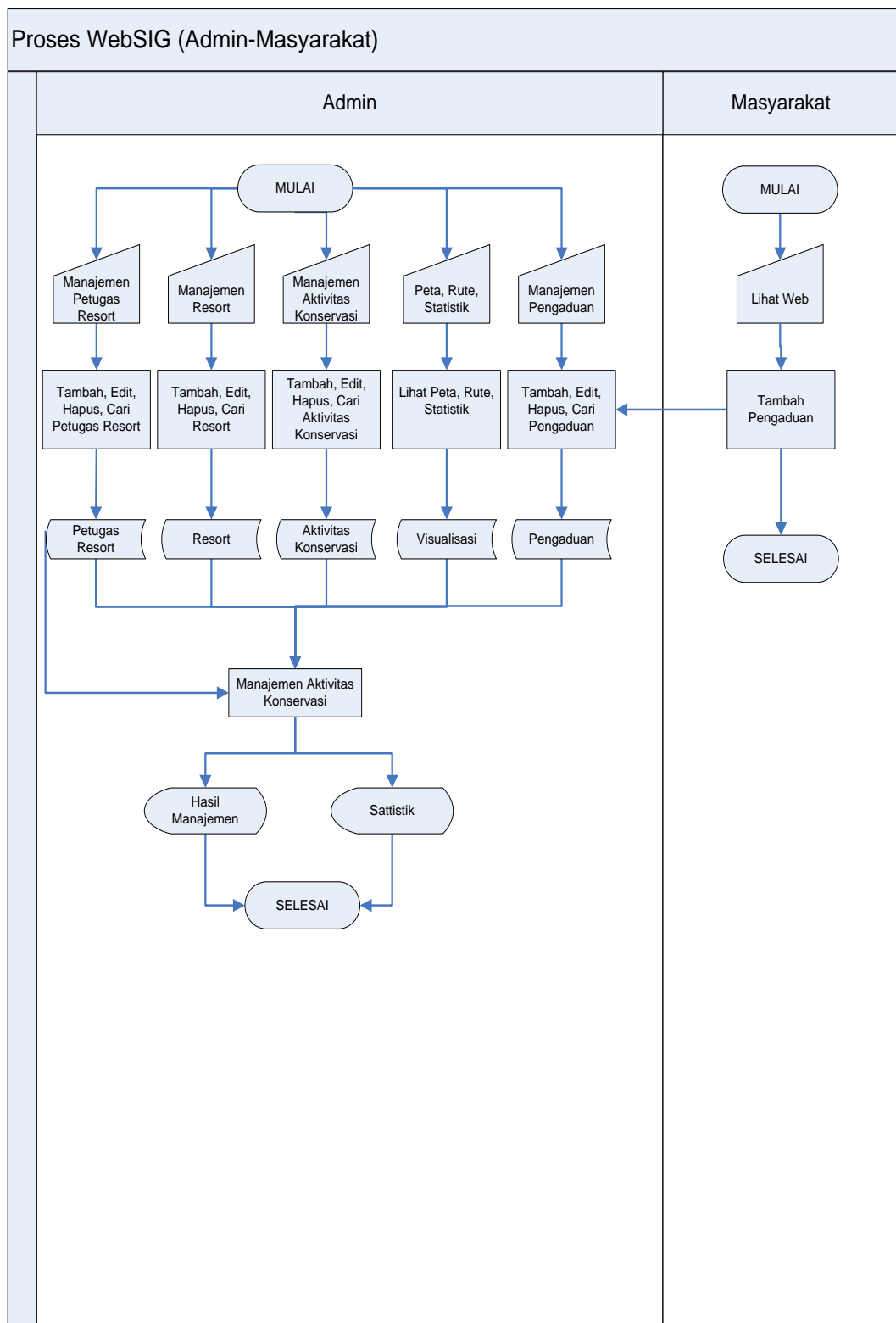


Gambar 3.3 Flowchart sistem admin dan petugas lapangan





Gambar 3.4 Flowchart sistem admin dan operator resort



Gambar 3.5 Flowchart sistem admin dan masyarakat

Bagan alir (*flowchart*) diatas meliputi keterkaitan setiap aktor terhadap sistem secara keseluruhan. Dalam penelitian ini terdapat empat aktor meliputi admin, *operator resort*, Polhut/petugas *resort* lapangan dan masyarakat. Proses dimulai dari petugas resort lapangan yang bertugas melakukan pengumpulan data secara langsung terhadap objek penelitian yaitu segala kegiatan konservasi seperti temuan flora, fauna, penebangan pohon ilegal, perusakan hutan dan lain-lain. Proses pengumpulan data oleh Polhut/petugas *resort* lapangan menggunakan aplikasi *Smartphone Android* dan terhubung langsung dengan web utama. Sehingga berbagai macam informasi yang terkait dengan kegiatan konservasi yang diberikan oleh Polhut/petugas *resort* lapangan akan terdeteksi dan dapat diambil tindakan. Kegiatan konservasi yang dilakukan oleh Polhut/petugas *resort* lapangan juga dapat dilakukan oleh *operator resort* yang terletak di kantor masing-masing *resort*, hal ini diperlukan apabila terdapat beberapa kendala dalam pengoperasian aplikasi *android*.

Selanjutnya, *operator resort* memiliki keterkaitan akses terhadap web utama yang dikelola oleh admin di Balai. Hal ini terlihat oleh kegiatan konservasi yang dilakukan baik dipihak *operator resort* maupun Polhut/petugas resort lapangan yang merujuk langsung ke web utama, khususnya bagian pengelolaan kegiatan konservasi. Web utama memiliki beberapa aktivitas meliputi manajemen Polhut/petugas *resort* dimana manajemen ini juga dapat dilaksanakan oleh *operator resort*, manajemen tata ruang *resort*, manajemen aktivitas konservasi, manajemen informasi dan manajemen pengaduan oleh masyarakat. Manajemen tersebut akan menghasilkan statistik hasil kegiatan konservasi.

Proses pengaduan oleh masyarakat diperlukan untuk mengantisipasi berbagai macam aktivitas konservasi yang luput dari pengawasan Polhut/petugas *resort* lapangan maupun Balai. Masyarakat dapat melakukan pengaduan di web utama, berisikan tentang apapun yang terjadi terkait dengan kejadian di kawasan maupun diluar *resort*.

### **3.6 Sarana Pendukung**

Dalam merancang dan membangun aplikasi ini dibutuhkan sarana pendukung berupa literatur-literatur, perangkat keras dan perangkat lunak.

1. Perangkat keras yang akan digunakan dalam penelitian ini berupa laptop Lenovo Thinkpad Edge 430 dengan spesifikasi sebagai berikut:

- 1) Prosesor Intel Core i5 3210M
- 2) Memori 4096 MB
- 3) Harddisk 700 GB, dengan *free space harddisk* 486 GB
- 4) VGA Intel(R) HD 4000
- 5) Monitor LCD 14" dengan resolusi 1366 x 768 *pixels*
- 6) *Keyboard* dan mouse
- 7) *Smartphone* Android 2.3 LG-P970
- 8) *GPS Tracker/Mobile GPS*
- 9) Printer Canon IP 2700 Series

2. Perangkat lunak yang digunakan dalam penelitian ini, antara lain:

- 1) Sistem Operasi: Windows 7 Ultimate
- 2) Android SDK dan ADT 8.0.1
- 3) Eclipse Indigo
- 4) SQLite
- 5) *Text editor* Notepad ++
- 6) Bahasa pemrograman: PHP version 5.2.6 dan JavaScript
- 7) SMD: MySQL version 5.0.51b
- 8) *Web browser* Mozilla Firefox 3.6

3. Literatur-literatur berupa buku (cetak maupun elektronik) dan jurnal ilmiah mengenai:

- 1) Sistem Informasi Geografis
- 2) Google API, *client-server* dan webgis.
- 3) Pemrograman web, Javascript, CSS dan HTML
- 4) Konservasi Flora, Fauna dan Hutan Provinsi Bengkulu

### **3.7 Jadwal Penelitian**

Penelitian mengenai Sistem Informasi Georafis Sebaran Flora, Fauna, Dan Aktivitas Konservasi Dalam Mendukung Kelestarian Alam Berbasis Manajemen Resort Provinsi Bengkulu dilaksanakan selama 6 bulan, terhitung dari bulan Maret-Agustus 2013. Adapun rincian waktu penelitian adalah sebagai berikut:

Tabel 3.1 Rencana Penelitian

No	Kegiatan	Tahun 2013					
		Maret	April	Mei	Juni	Juli	Agustus
1	Penyusunan proposal & Pembuatan peta kerja ( <i>Time Schedule</i> )						
2	Pengumpulan data Resort						
3	Analisis hasil pengumpulan data sekunder dan primer di BKSDA dan DISHUT Prov. Bengkulu						
4	Implementasi sistem berbasis WEB SIG dengan Google Maps API						
5	Penulisan laporan Penelitian						
6	Evaluasi dan Presentasi Hasil Penelitian						

## **BAB IV**

### **ANALISIS DAN PERANCANGAN SISTEM**

Dalam penelitian ini model pengembangan sistem yang digunakan adalah sequensial linier atau model *waterfall* (Sommerville, 2003). Model ini merupakan pendekatan pengembangan perangkat lunak sistematis yang dimulai dari analisis, desain, penulisan program, pengujian dan pemeliharaan. Adapun tahapan-tahapan pengembangan sistem yang dilakukan adalah (1) *Requirements analysis and Definition* (Definisi dan Analisis Kebutuhan), (2) *System and Software Design* (Desain Sistem dan Perangkat Lunak), (3) *Implementation and Unit Testing* (Implementasi dan Pengujian Unit), (4) *Integration and System Testing* (Integrasi dan Pengujian Sistem), dan (5) *Operation and Maintenance* (Operasi dan Pemeliharaan). Tahapan (1) dan (2) akan dibahas pada bab ini sedangkan tahapan (3), (4) dan (5) akan dibahas pada bab selanjutnya.

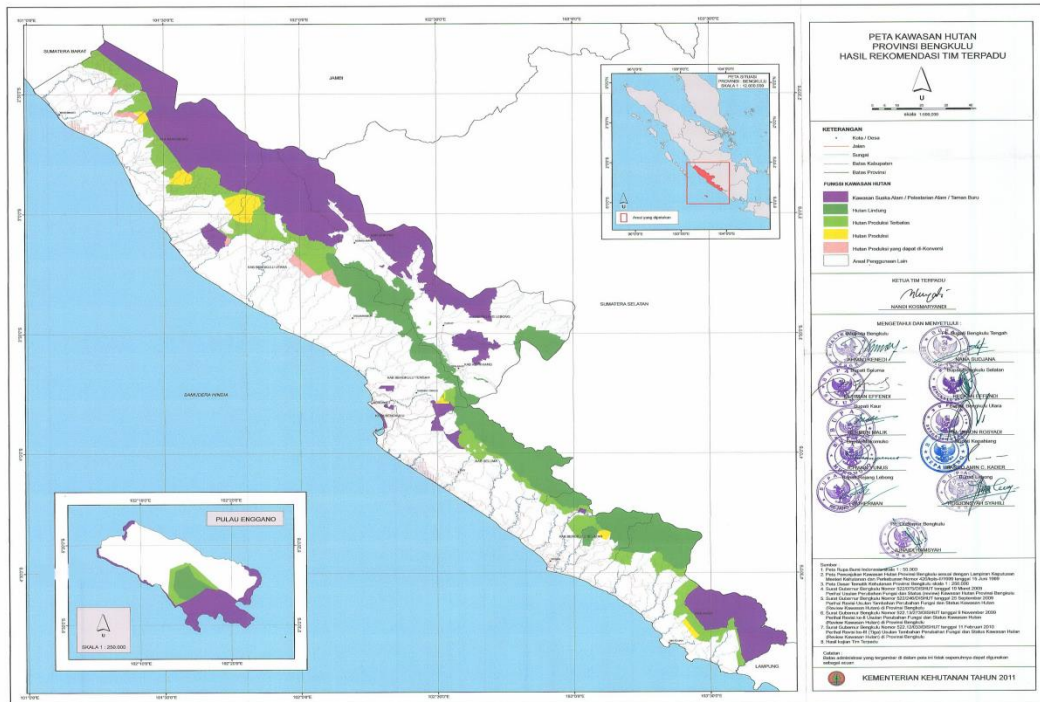
#### **4.1 Definisi dan Analisis Kebutuhan**

##### **4.1.1 Identifikasi Masalah**

Sebagaimana telah dijelaskan pada rumusan masalah, belum tersedianya aplikasi untuk mendukung kegiatan konservasi maka perlu diciptakannya sebuah aplikasi baru yang memanfaatkan fungsi geografis yang dapat membantu dan menjembatani lalu lintas informasi terkait dengan masalah tersebut sehingga dapat bermanfaat bagi masyarakat, instansi maupun pihak-pihak yang terkait.

##### **4.1.2 Analisis Sistem Yang Ada**

Berdasarkan hasil observasi di Balai Konservasi dan Sumber Daya Alam dan Dinas Kehutanan Provinsi Bengkulu terkumpul beberapa informasi terkait penelitian. Salah satunya terlihat dari peta tematik kawasan hutan Provinsi Bengkulu meliputi kawasan konservasi beserta pembagian wilayahnya.



Gambar 4.1 Peta Kawasan Hutan meliputi kawasan Konservasi

(Sumber : Dinas Kehutanan Provinsi Bengkulu, 2013)

Berdasarkan hal tersebut, dibutuhkan suatu aplikasi dan sistem baru yang mampu mendukung akomodasi informasi terkait kegiatan konservasi tidak hanya dalam bentuk peta tetapi juga mampu menjembatani laju informasi yang lebih dinamis dan fleksibel. Adapun permasalahan yang terjadi sebagai berikut :

1. Sistem yang ada saat ini masih sangat terbatas kegiatan konservasi di masing-masing resort yang tersebar di 12 daerah di provinsi Bengkulu masih dilakukan dengan cara konvensional atau manual. Data kegiatan konservasi terpusat pada masing-masing resort dan tidak berbasis web. Hal tersebut berdampak pada lambatnya arus informasi dan kecenderungan untuk memunculkan manipulative data. Selanjutnya data yang ditampilkan masih sangat terbatas dan cenderung terpaku dengan “teks” dan belum terkomputerisasi dengan baik.
2. Kurangnya pengetahuan masyarakat mengenai kawasan konservasi dan kegiatan konservasi. Sehingga seringkali terjadi pelanggaran karena ketidakjelasan hal tersebut. Perlu adanya suatu sistem yang menangani maupun memberikan informasi terkait.

3. Aplikasi dan sistem yang mendukung kegiatan konservasi merupakan salah satu cita-cita BKSDA maupun DISHUT Provinsi Bengkulu dalam memanajemen kawasan, maupun kegiatan konservasi di masing-masing *resort*.

#### **4.1.3 Analisis Kebutuhan Pengguna**

Dari data yang diperoleh melalui wawancara Polhut(Polisi Hutan) BKSDA, informan PIKA dan studi pustaka meliputi pengumpulan informasi terkait kegiatan konservasi, maka didapat beberapa kebutuhan pengguna yang diharapkan dapat dipenuhi meliputi :

1. Admin dan operator Web bertugas melakukan manajemen *resort*, manajemen petugas *resort* dan manajemen kegiatan konservasi serta memberikan gambaran umum kegiatan konservasi dalam bentuk peta, rute, dan grafik serta informasi pengaduan dari masyarakat.
2. Pengguna dapat melakukan penelusuran informasi terkait dengan kegiatan konservasi, peta *resort*, peta aktivitas konservasi, galeri kegiatan serta pengecekan kawasan *resort* pada.

Selanjutnya dilakukanlah pengumpulan data yang terkait dengan kebutuhan pengguna seperti informasi kawasan konservasi di Provinsi Bengkulu, kegiatan konservasi yang sedang dilaksanakan serta aksesibilitas dari dan menuju kawasan konservasi (Insitu) maupun yang telah diluar wilayah konservasi (Eksitu). Adapun pengumpulan data yang biasa dilakukan dalam kegiatan survei lapangan meliputi *tracking* menggunakan alat GPS maupun *mobile* GPS dan data dari Polhut/petugas *resort* yang nantinya akan diintegrasikan ke peta web.

#### **4.1.4 Analisis Kebutuhan Sistem**

Berdasarkan hasil penelitian yang telah dilakukan maka kebutuhan sistem yang diperlukan dalam pembuatan aplikasi android terintegrasi web terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*). Bahan utama yang digunakan adalah peta tematik kawasan resort Provinsi Bengkulu serta peta digital hasil integrasi Google Maps API. Data kegiatan konservasi didapat dari BKSDA dan DISHUT provinsi Bengkulu, selanjutnya informasi terkait mengenai



dokumentasi foto kegiatan konservasi dan penentuan posisi berdasarkan GPS pada kawasan konservasi. Adapun perangkat keras (*hardware*) dan perangkat lunak (*software*) yang digunakan pada penelitian ini meliputi :

1. Perangkat keras yang digunakan dalam penelitian ini, antara lain:

- 10) Prosesor Intel Core i5 3210M
- 11) Memori 4096 MB
- 12) Harddisk 700 GB, dengan *free space harddisk* 486 GB
- 13) VGA Intel(R) HD 4000
- 14) Monitor LCD 14" dengan resolusi 1366 x 768 *pixels*
- 15) *Keyboard* dan mouse
- 16) *Smartphone* Android 2.3 LG-P970
- 17) *GPS Tracker/Mobile GPS*
- 18) Printer Canon IP 2700 Series

2. Perangkat lunak yang digunakan dalam penelitian ini, antara lain:

- 9) Sistem Operasi: Windows 7 Ultimate
- 10) Eclipse Indigo dan SQLite
- 11) *Text editor* Notepad ++
- 12) Bahasa pemrograman: PHP version 5.2.6 dan JavaScript
- 13) SMD: MySQL version 5.0.51b
- 14) *Web browser* Mozilla Firefox 3.6

3. Literatur-literatur berupa buku (cetak maupun elektronik) dan jurnal ilmiah mengenai:

- 5) Pemrograman Android
- 6) Sistem Informasi Geografis
- 7) Google API V.3
- 8) Pemrograman web, Javascript, CSS dan HTML
- 9) Kegiatan konservasi di Provinsi Bengkulu

## 4.2 Desain Sistem dan Perangkat Lunak

### 4.2.1 Perancangan Model UML

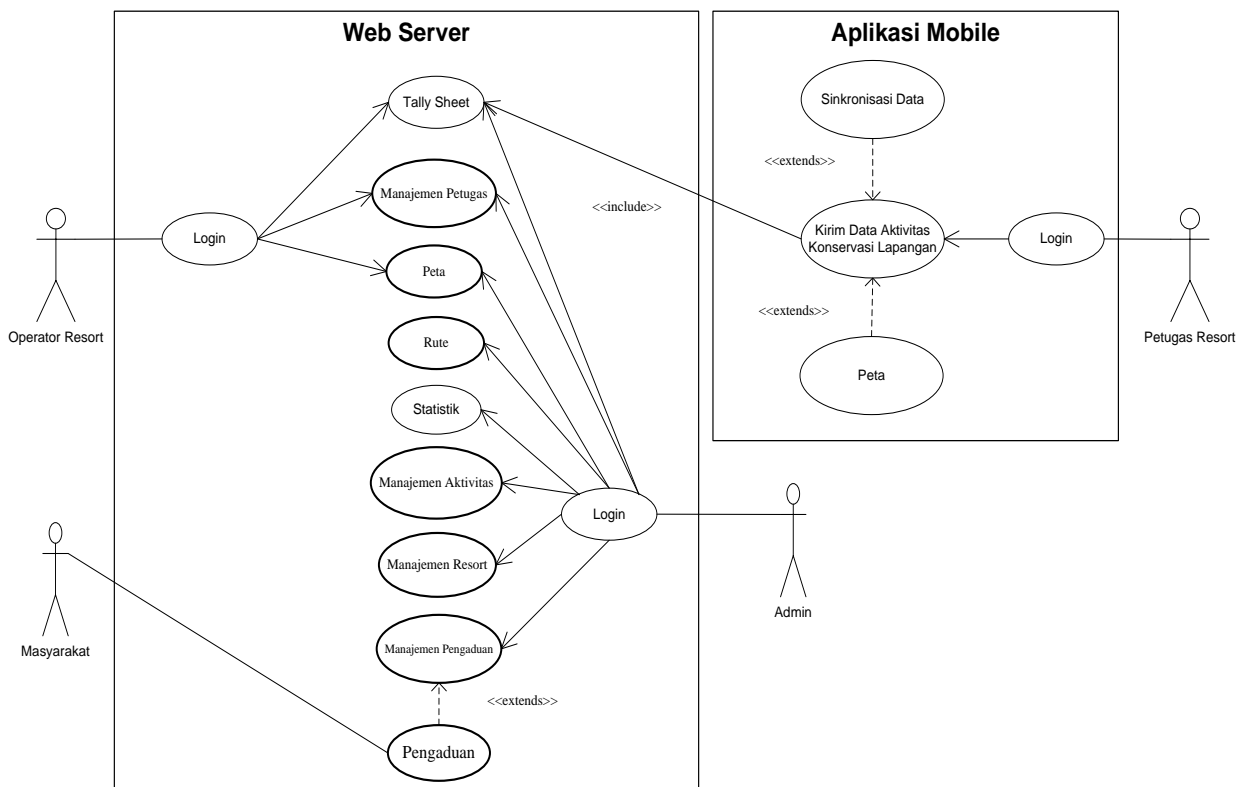
Pada perancangan dengan model UML (*Unified Modelling Language*) 2.0 dibagi menjadi dua kelompok utama yaitu pemodelan sifat (*behavior diagrams*) dan pemodelan struktur (*structure diagrams*) dari sistem.

#### 4.2.1.1 Behaviour Diagrams

Berikut diagram-diagram pemodelan sifat yang digunakan dalam merancang sistem ini :

#### 3. Use Case Diagram

*Use case diagram* merupakan gambaran interaksi antara pengguna sistem (*actor*) dengan kejadian atau kasus (*use case*) yang disesuaikan dengan langkah-langkah (*scenario*) yang telah ditentukan. Aktor menggambarkan orang maupun sistem atau *external entitas/stakeholder* yang menyediakan atau menerima informasi dari sistem. Berikut *use case diagram* dari sistem yang akan dibangun :

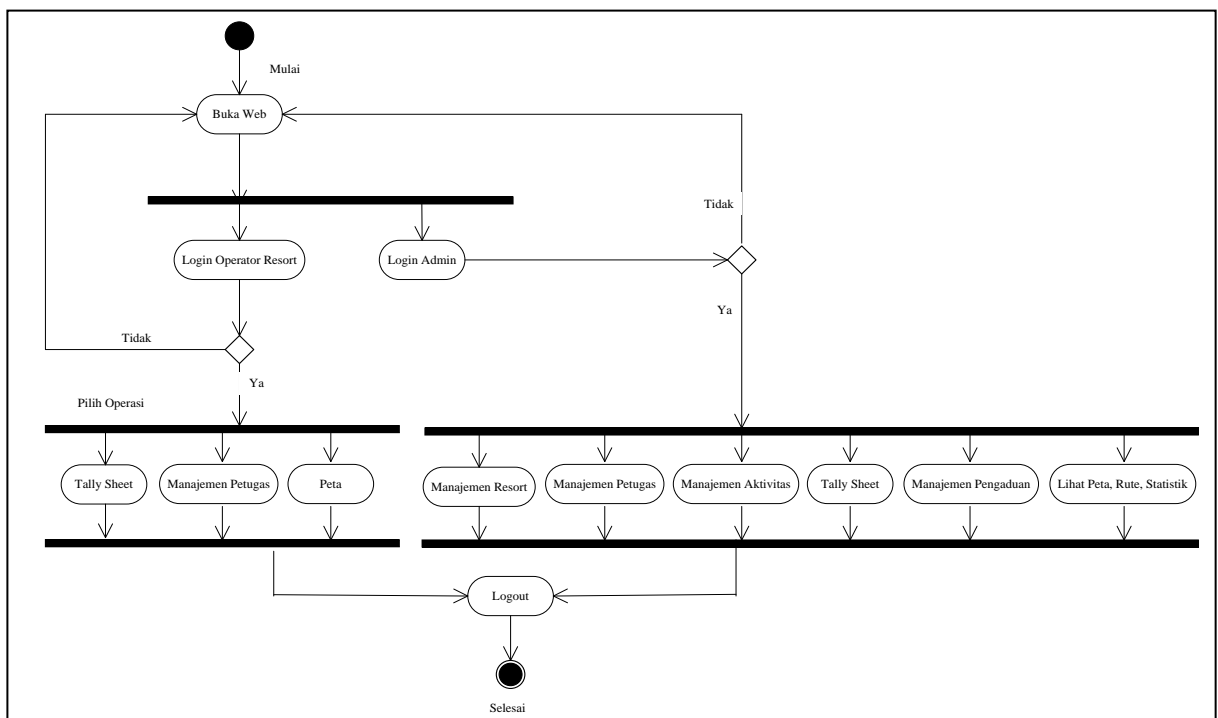


Gambar 4.2 Use Case Diagram

Terlihat pada *use case diagram* diatas, terdapat empat aktor yang terlibat langsung pada sistem meliputi admin, *operator resort*, petugas *resort* lapangan dan masyarakat. Admin memiliki kuasa penuh dalam mengelola sistem secara keseluruhan, hal tersebut dilihat dari akses admin di semua manajemen meliputi manajemen *resort*, manajemen petugas *resort*, *tally sheet*, manajemen aktivitas, lihat peta, rute, statistik dan manajemen pengaduan. Sedangkan *operator resort* hanya mampu melakukan penambahan *tally sheet*, lihat peta dan manajemen petugas *resort* lapangan. Begitupun petugas *resort* lapangan yang bertindak langsung dalam penambahan aktivitas konservasi melalui aplikasi *mobile*. Sedangkan masyarakat mampu memberikan pengaduan secara langsung terkait dengan aktivitas konservasi.

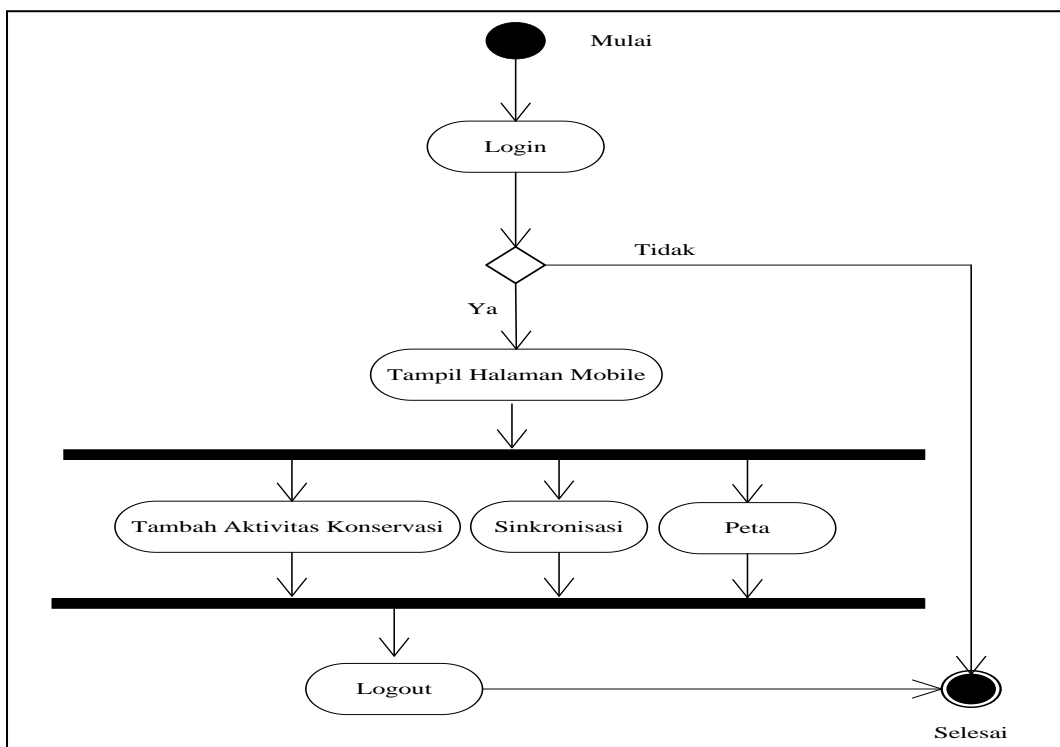
#### 4. Activity Diagram

*Activity Diagram* merupakan representasi dari alur kerja tahapan aktivitas. Diagram ini mendukung pilihan tindakan, iterasi dan *concurrency*. *Activity diagram* dapat digunakan sebagai penjelasan bisnis dan alur kerja operasional secara *step by step* dari komponen suatu sistem. *Activity diagram* menunjukkan keseluruhan dari aliran kontrol. Berikut *activity diagram* dari sistem yang akan dibuat :



Gambar 4.3 Activity Diagram Web

Terlihat pada *Activity* diagram Web diatas, aktivitas dimulai dengan mengakses web terlebih dahulu, selanjutnya autentifikasi akses sebagai admin atau operator resort. Admin memiliki kuasa penuh dalam mengelola sistem secara keseluruhan, hal tersebut dilihat dari akses admin di semua manajemen meliputi manajemen *resort*, manajemen petugas *resort*, *tally sheet*, manajemen aktivitas, lihat peta, rute, statistik dan manajemen pengaduan. Sedangkan operator *resort* hanya mampu melakukan penambahan *tally sheet*, lihat peta dan manajemen petugas resort lapangan. Setelah melakukan akses masing-masing aktivitas ditutup dengan *logout* dan aktivitas selesai.



Gambar 4.4 *Activity Diagram Aplikasi Mobile*

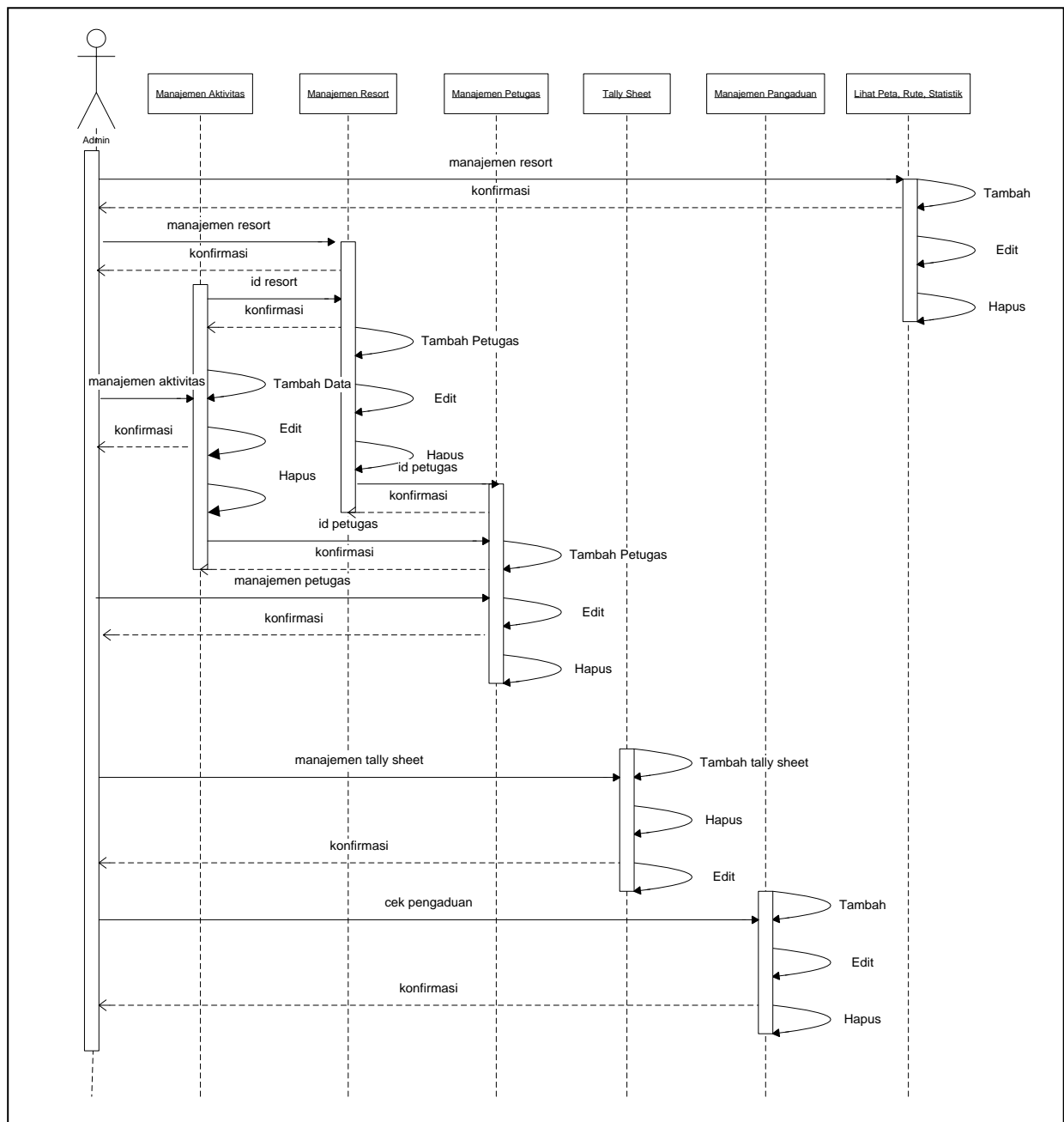
Activity Diagram Aplikasi Mobile diatas menunjukkan urutan kegiatan aktivitas konservasi yang dilakukan melalui aplikasi oleh petugas resort lapangan. Dimulai dengan login, mengakses halaman mobile, dan melakukan pilihan aktivitas meliputi tambah aktivitas konservasi, sinkronisasi dan peta. Selanjutnya aktivitas ditutup dengan logout dan selesai.

## 5. Interaction Diagram

Interaction Diagram terdiri dari empat diagram yaitu :

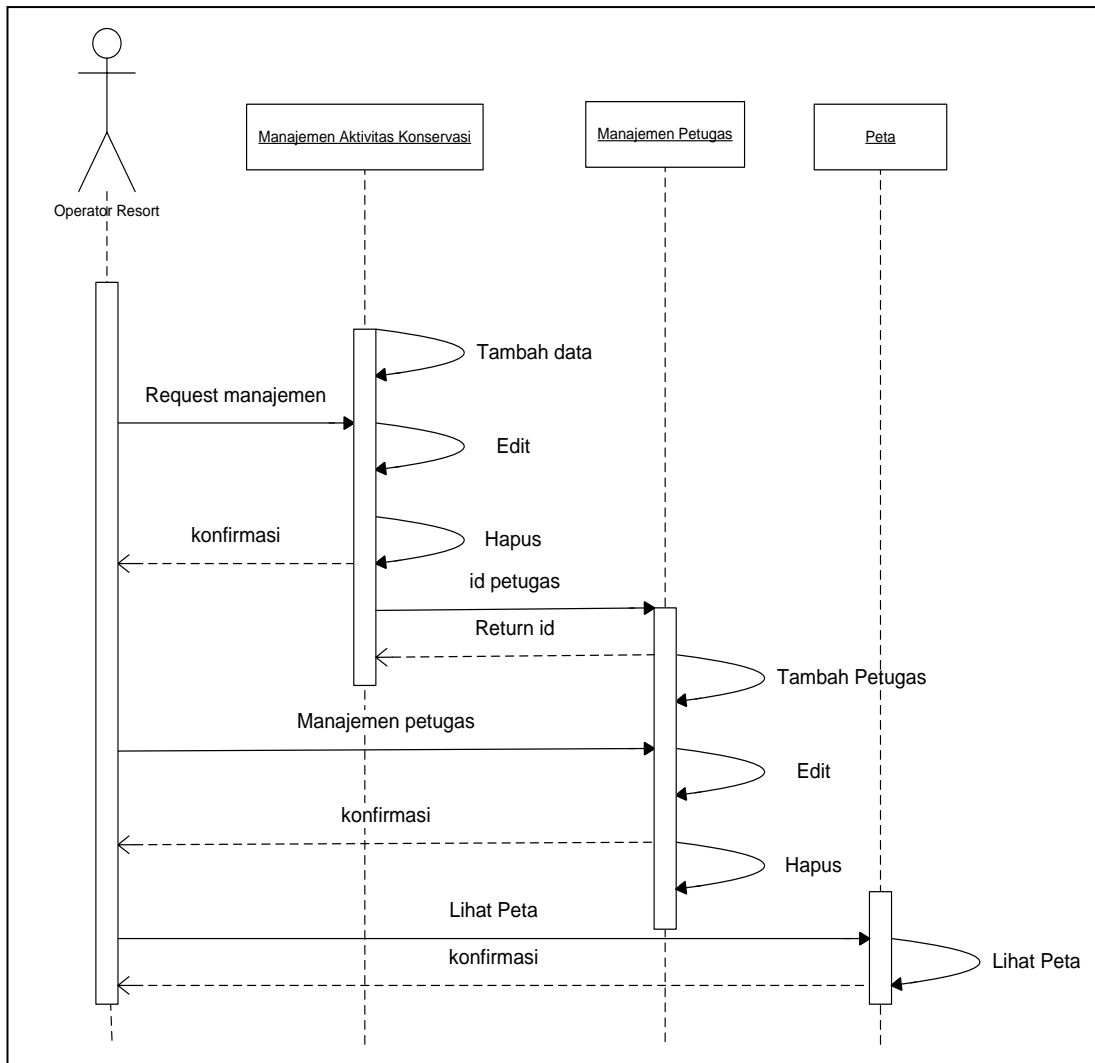
### 1) Sequence Diagram

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu. Berikut *sequence diagram* dari sistem yang akan dibuat :



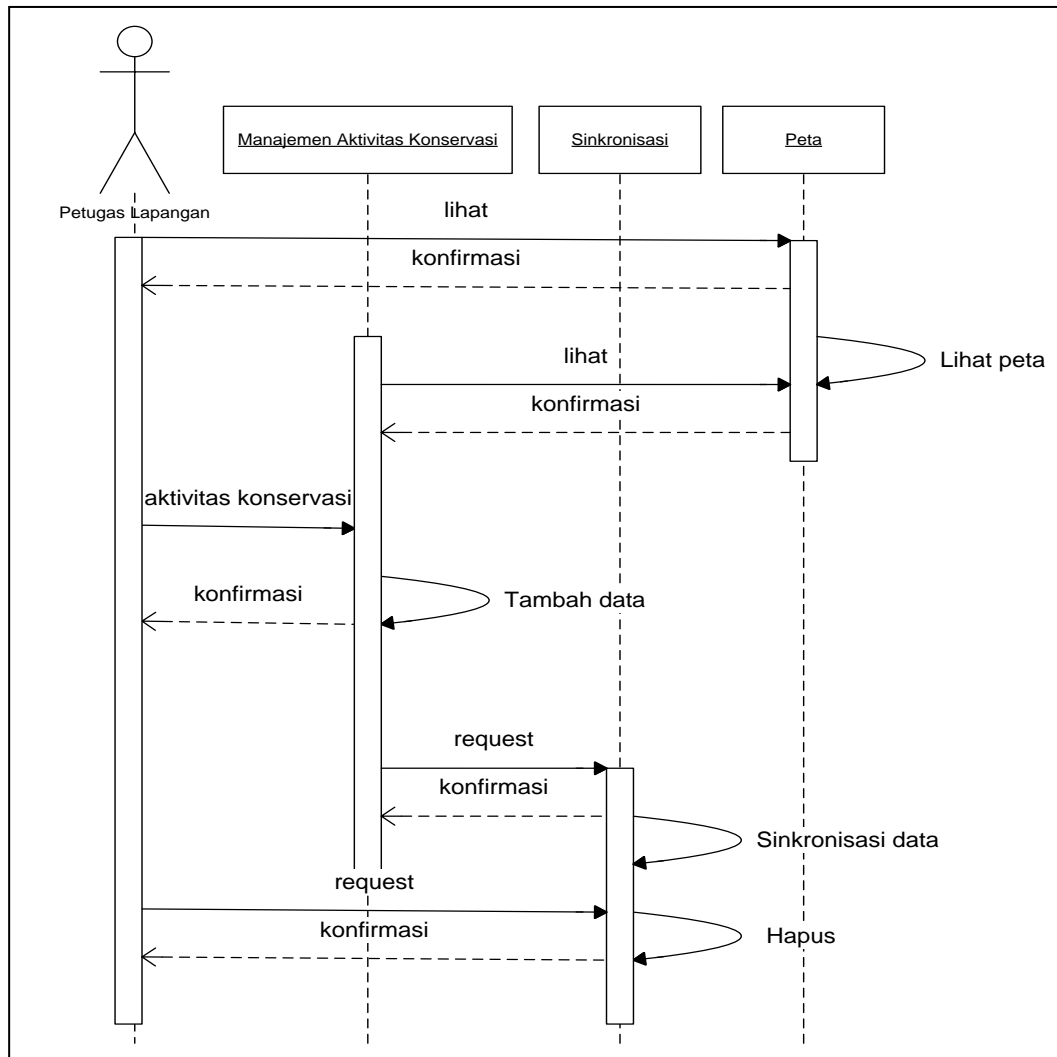
Gambar 4.5 Sequence Diagram Web

Pada *sequence diagram* diatas, terdapat enam manajemen yang berhubungan oleh admin selaku aktor meliputi manajemen resort, manajemen petugas, manajemen aktivitas, tally sheet, lihat peta, rute, statistik dan manajemen pengaduan. Pada masing-masing manajemen, admin memiliki akses untuk menambah, manghapus maupun mengubah data berdasarkan urutan dan waktunya.



Gambar 4.6 *Sequence Diagram Manajemen Konservasi dan Petugas*

Terlihat pada *sequence diagram* kedua diatas, terdapat tiga aktivitas yang berhubungan oleh operator selaku aktor meliputi manajemen petugas, lihat peta dan manajemen aktivitas konservasi. Pada masing-masing manajemen, operator memiliki akses untuk menambah, manghapus maupun mengubah data berdasarkan urutan dan waktunya.

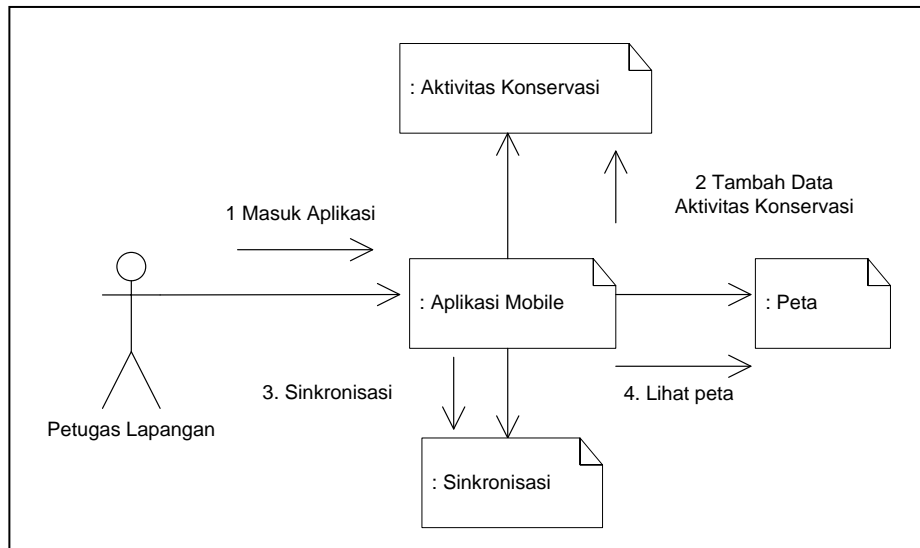


Gambar 4.7 *Sequence Diagram* Aktivitas Konservasi

Pada *sequence diagram* ketiga diatas, menunjukan aktivitas petugas *resort* lapangan. Terdapat tiga manajemen yang berhubungan dengan petugas *resort* lapangan selaku aktor meliputi tambah aktivitas konservasi, lihat peta dan sinkronisasi. Terdapat fungsi kirim, sinkronisasi, dan hapus serta fungsi lihat peta pada diagram di atas.

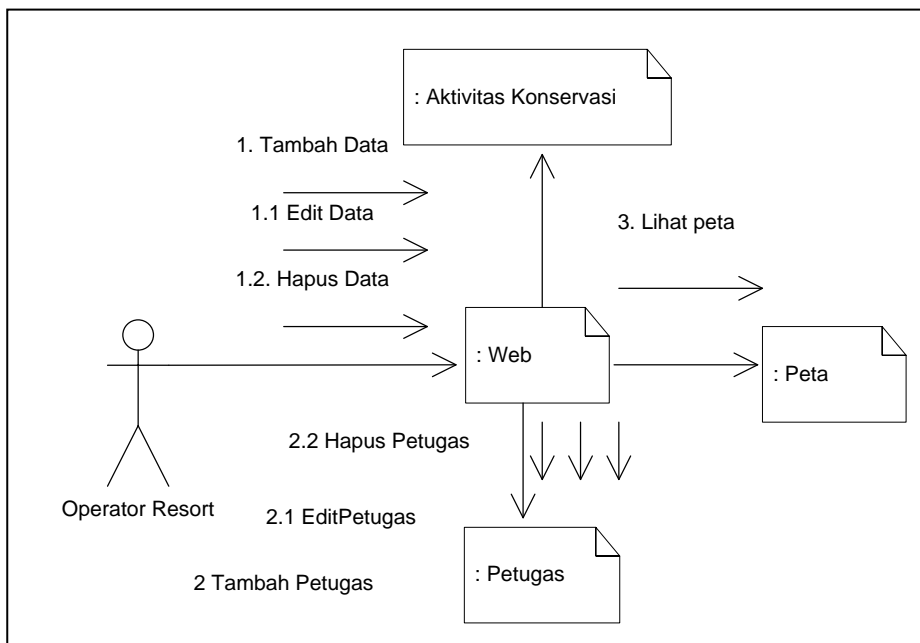
## 2) *Communication Diagram*

*Communication Diagram* merupakan diagram yang memfokuskan pada komunikasi yang berhubungan dengan struktur dari objek yang terlibat dalam suatu tugas. Berikut *Communication Diagram* dari sistem yang akan dibuat :



Gambar 4.8 *Communication Diagram Aplikasi*

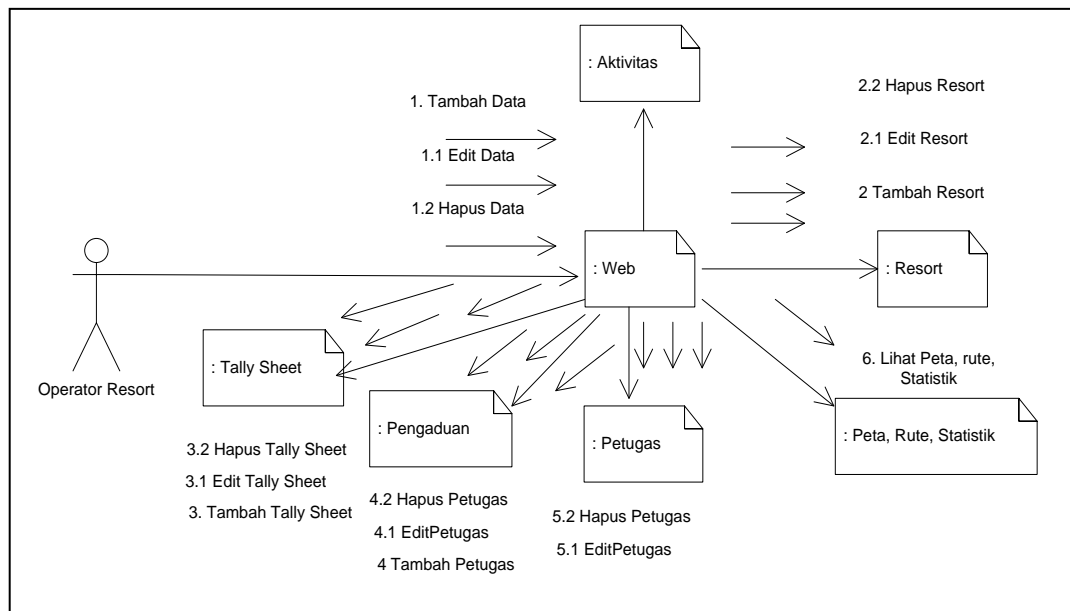
Terlihat pada *communication diagram* diatas terdapat komunikasi antara petugas *resort* lapangan aplikasi *mobile* yang digunakan untuk menjalankan aktivitas konservasi, melakukan sinkronisasi data dan melihat peta.



Gambar 4.9 *Communication Diagram Operator Resort*

Pada *communication diagram* diatas terlihat komunikasi antara oprator *resort* dengan web. Komunikasi meliputi penambahan, penghapusan dan pengubahan aktivitas konservasi dan petugas resort serta lihat peta oleh operator *resort*.



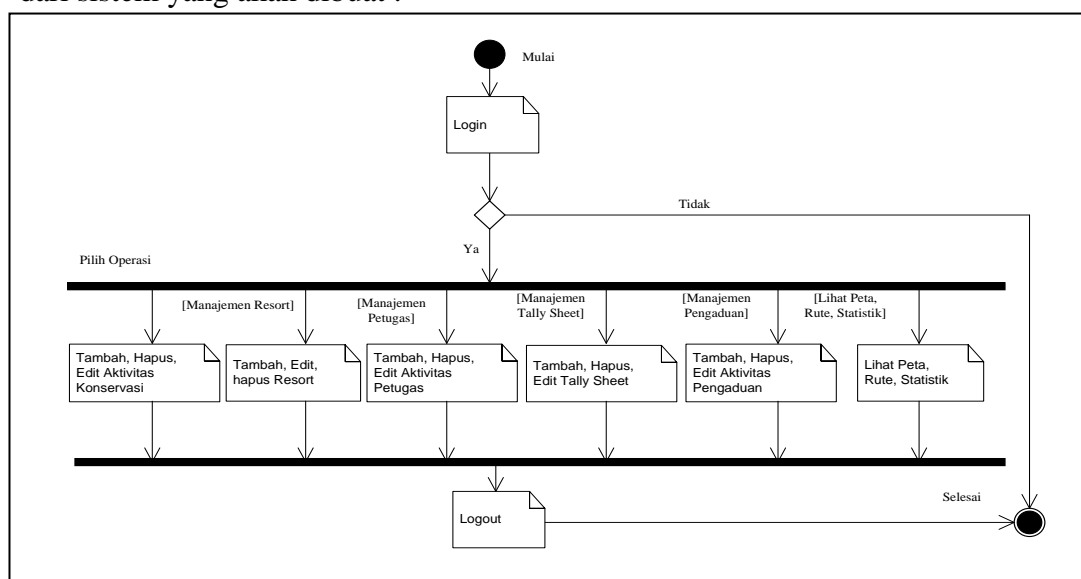


Gambar 4.10 *Communication Diagram Admin*

Selanjutnya pada *communication diagram* ketiga diatas terdapat komunikasi antara admin dengan wab yang dikelola. Komunikasi antara admin dan web meliputi aktivitas konservasi, resort, petugas, berita, lihat peta, rute, statistik dan pengaduan. Komunikasi admin juga berlaku untuk penambahan, penghapusan dan perubahan masing-masing objek.

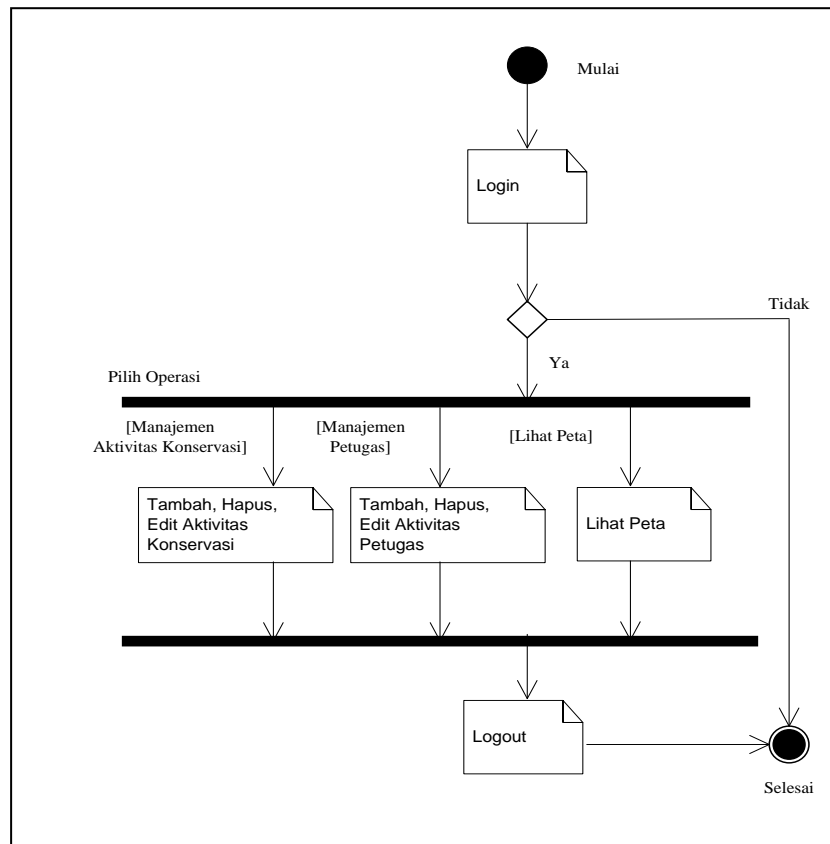
### 3) *Interaction Overview Diagram*

*Interaction Overview Diagram* berfokus pada gambaran aliran kontrol interaksi. *Interaction Overview Diagram* menggambarkan interaksi di mana pesan dan *lifelines* tersembunyi. Berikut *Interaction Overview Diagram* dari sistem yang akan dibuat :



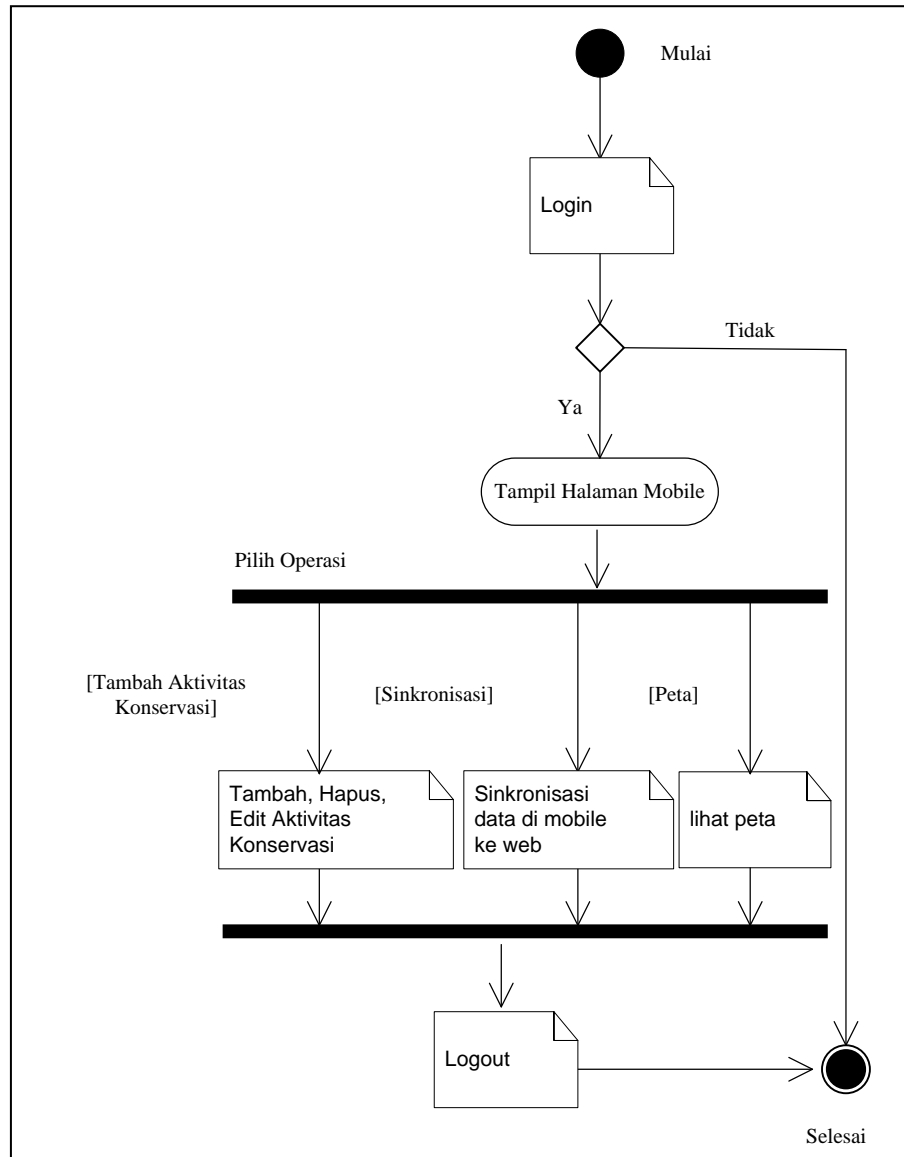
Gambar 4.11 *Interaction Overview Diagram Admin*

Terlihat pada *Interaction Overview Diagram* diatas, hampir sama seperti *activity diagram*, pada diagram ini lebih diperjelas mengenai aksi apa yang dilakukan. Interaksi dimulai dengan mengakses web terlebih dahulu. Admin memiliki kuasa penuh dalam mengelola sistem secara keseluruhan, hal tersebut dilihat dari akses admin di semua manajemen meliputi manajemen *resort*, manajemen petugas *resort*, manajemen aktivitas konservasi, manajemen informasi, lihat peta, rute, statistic, dan manajemen pengaduan. Setelah melakukan akses aktivitas ditutup dengan *logout* dan aktivitas selesai.



Gambar 4.12 *Interaction Overview Diagram Operator*

Hampir sama dengan *activity diagram* sebelumnya, *Interaction Overview Diagram* diatas lebih diperinci mengenai aksi yang dilakukan. Diagram dimulai dengan mengakses web terlebih dahulu. Operator memiliki kuasa mengelola sistem manajemen petugas *resort*, lihat peta dan manajemen aktivitas konservasi. Setelah melakukan akses aktivitas ditutup dengan *logout* dan aktivitas selesai.



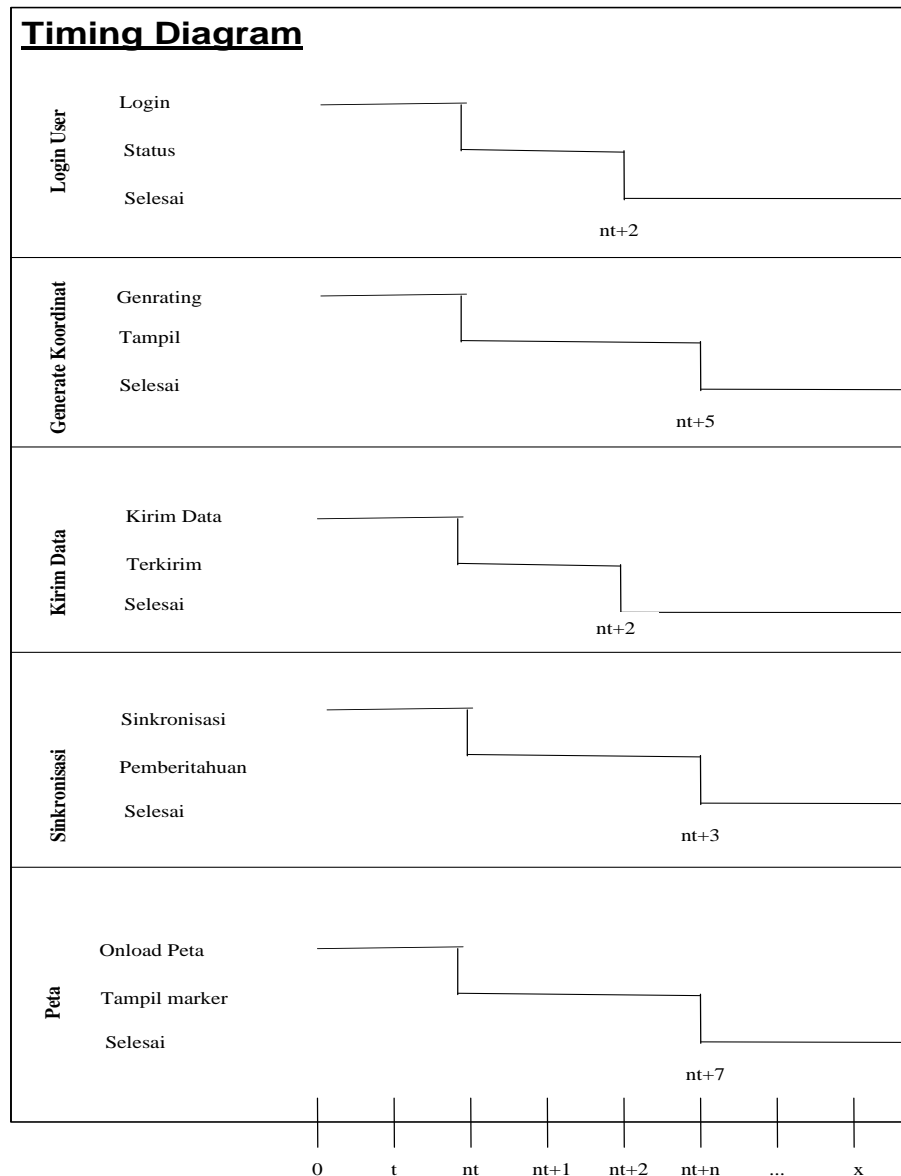
Gambar 4.13 *Interaction Overview Diagram Petugas*

*Interaction Overview Diagram* diatas lebih diperinci mengenai aksi petugas *resort* lapangan terhadap aplikasi *mobile*. Diagram dimulai dengan *login* aplikasi terlebih dahulu. Petugas *resort* lapangan melakukan interaksi terhadap aplikasi meliputi aktivitas konservasi, lihat peta dan sinkronisasi data. Setelah melakukan akses aktivitas ditutup dengan *logout* dan aktivitas selesai.

#### 4) *Timing Diagram*

*Timing Diagram* digunakan untuk mengeksplorasi perilaku benda sepanjang jangka waktu tertentu. Sebuah *Timing Diagram* adalah bentuk khusus dari *sequence diagram*. Perbedaan antara *timing diagram* dan *sequence diagram* adalah

sumbu dibalik sehingga waktu meningkat dari kiri ke kanan dan jalur hidup ditunjukkan dalam kompartemen terpisah yang tersusun vertikal. Berikut *Timing Diagram* dari sistem yang akan dibuat :



Gambar 4.14 *Timing Diagram*

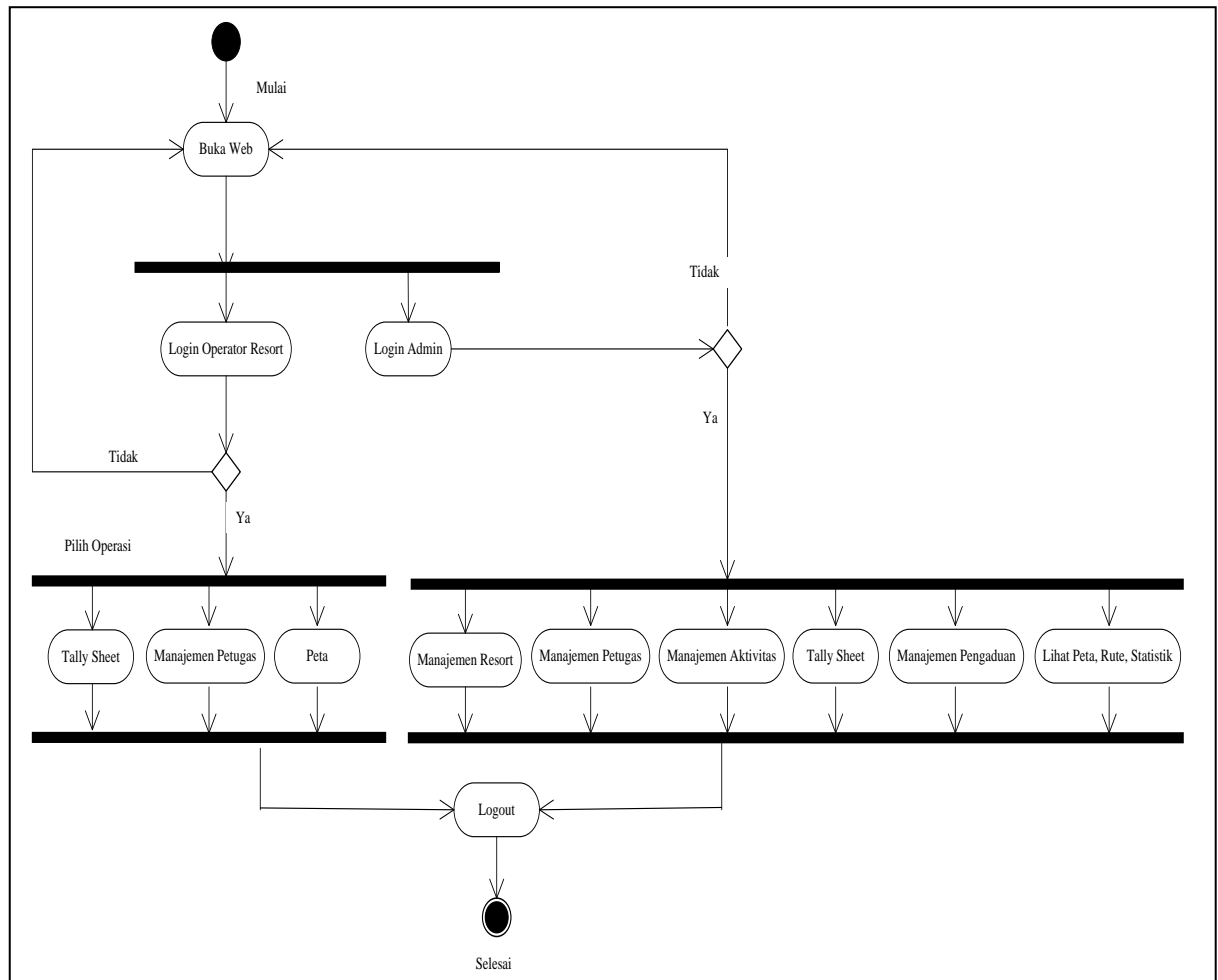
## 6. State Machine Diagram

Diagram ini terdiri dari dua diagram :

### 1) Behavior State Machine Diagram

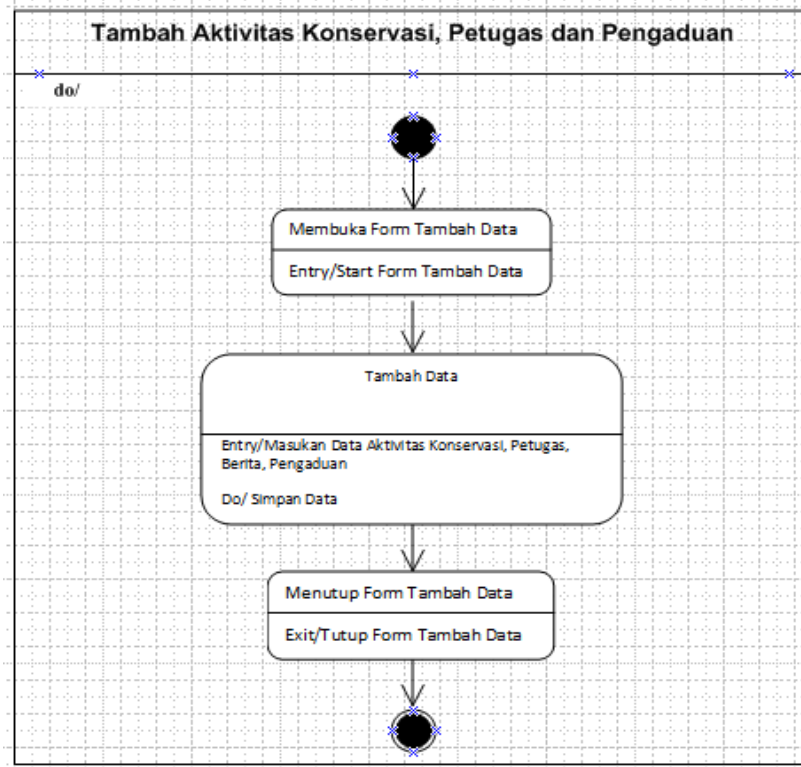
*Behavior State Machine Diagram* merupakan spesialisasi perilaku dan digunakan untuk menentukan perilaku diskrit bagian dari sistem yang dirancang melalui *state transitions* yang terbatas. Formalisme *state machine* yang digunakan

dalam hal ini adalah varian berbasis obyek *statecharts* Harel. Berikut *Behavior State Machine Diagram* dari sistem yang akan dibuat :



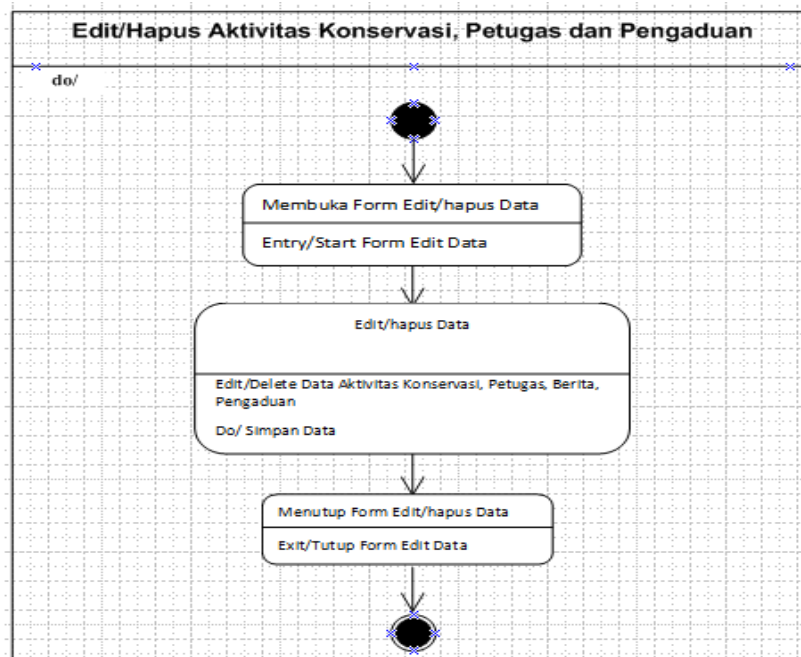
Gambar 4.15 *Behavior State Machine Diagram*

Pada *Behavior State Machine Diagram* diatas, admin memiliki kuasa penuh dalam mengelola sistem secara keseluruhan, hal tersebut dilihat dari perilaku admin di semua manajemen meliputi manajemen *resort*, manajemen petugas *resort*, manajemen aktivitas konservasi, lihat peta, rute, statistik dan manajemen pengaduan. Sedangkan operator *resort* hanya mampu melakukan penambahan aktivitas konservasi dan manajemen petugas resort lapangan.



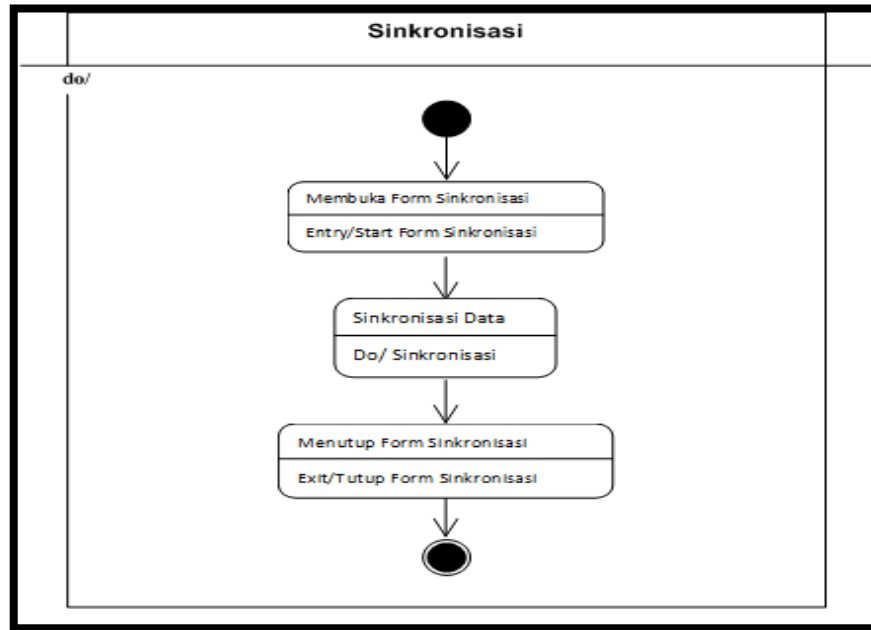
Gambar 4.16 Sub State Tambah Data

Pada *substate* tambah data diatas terlihat spesialisasi perilaku yang ditunjukkan dalam proses ini, dimulai dari membuka form tambah data, melakukan tambah data serta *entry* data dan menutup *form*.



Gambar 4.17 Sub State Edit Data

Pada *substate* edit dan hapus data diatas terlihat spesialisasi perilaku yang ditunjukkan dalam proses ini, dimulai dari membuka form edit dan hapus data, melakukan edit dan hapus dan menutup *form*.

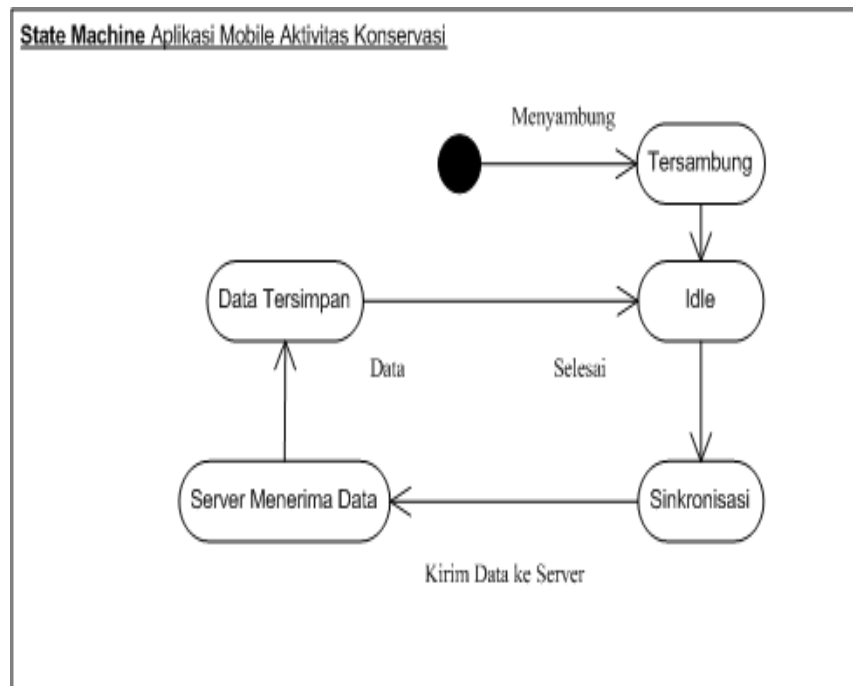


Gambar 4.18 Sub State Sinkronisasi Data

Pada *substate* sinkronisasi data juga terlihat spesialisasi perilaku yang ditunjukkan dalam proses ini, dimulai dari membuka form sinkronikasi data, melakukan sinkronikasi dan menutup *form* sinkronikasi.

## 2) Protocol State Machine Diagram

*Protocol State Machine Diagram* adalah spesialisasi *behavioral state machine* dan digunakan untuk mengekspresikan protokol penggunaan atau siklus hidup *classifier*. Ini menentukan operasi *classifier* mana yang dapat dipanggil dan dalam kondisi apa, sehingga menentukan urutan panggilan yang diperbolehkan pada operasi *classifier* ini. *Protocol State Machine Diagram* mengekspresikan transisi legal yang *classifier* dapat picu. Berikut *Protocol State Machine Diagram* dari sistem yang akan dibuat :



Gambar 4.19 *Protocol State Machine Diagram*

Pada *Protocol State Machine Diagram* diatas terlihat protokol penggunaan aplikasi *mobile* aktivitas konservasi, dimulai dengan urutan menyambung diawal *protocol* penggunaan, dilanjutkan dengan *idle* dan aktivitas sinkronisasi data yang ditujukan ke server, setelah data tersimpan, maka protokol kembali ke *idle*.

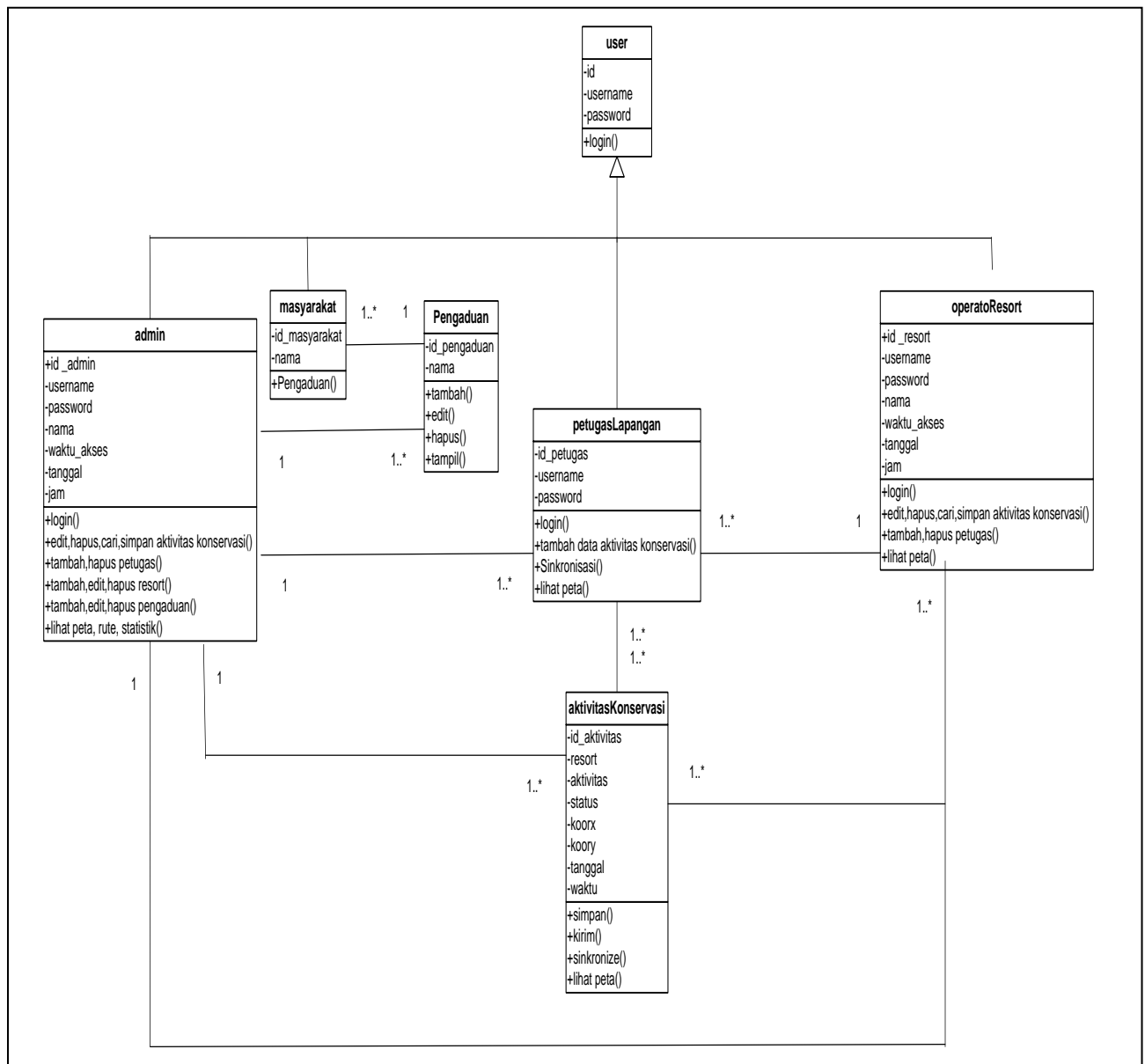
#### 4.2.1.2 *Structure Diagrams*

Berikut diagram pemodelan struktur yang digunakan untuk merancang sistem ini :

##### 1. *Class Diagram*

*Class Diagram* adalah diagram yang menunjukkan *class-class* yang ada dari sebuah sistem dan hubungannya secara logika. *Class diagram* menggambarkan struktur statis dari sebuah sistem. Karena itu class diagram merupakan tulang punggung atau kekuatan dasar dari hampir setiap metode berorientasi objek termasuk UML (Henderi, 2008). Sementara menurut ahli *class diagram* adalah gambar grafis mengenai struktur objek statis dari suatu sistem, menunjukkan *class-class* objek yang menyusun sebuah sistem dan juga hubungan antara *class* objek tersebut (Whitten L. Jeffery et al, 2004).



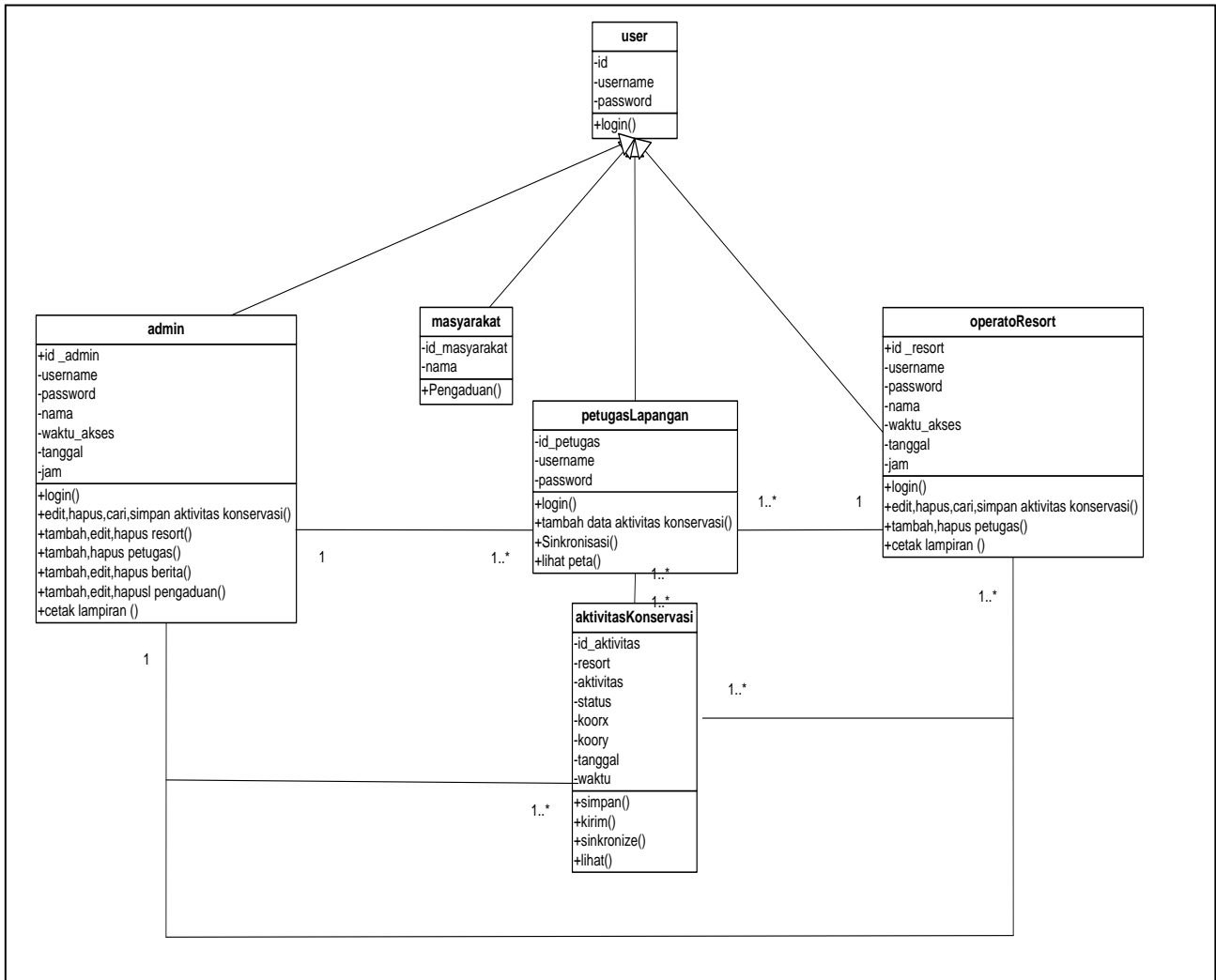


Gambar 4.20 *Class Diagram*

Terlihat pada *class diagram* diatas terdapat beberapa kelas yang saling terhubung satu dengan lainnya. Khusus untuk kelas admin, operator *resort*, masyarakat dan petugas *resort* lapangan digeneralisasikan menjadi kelas *user*. Selanjutnya, masing-masing kelas memiliki keterkaitannya masing-masing dengan kelas lain seperti kelas admin dengan kelas masyarakat, petugas *resort* lapangan, operator *resort*, aktivitas konservasi dan pengaduan. Begitu juga dengan kelas pengaduan dan kelas masyarakat, serta kelas operator *resort* dengan kelas aktivitas konservasi dan kelas petugas lapangan.

## 2. Object Diagram

*Object diagram* adalah diagram yang menunjukkan pandangan lengkap atau sebagian dari struktur sistem yang dimodelkan pada waktu tertentu.

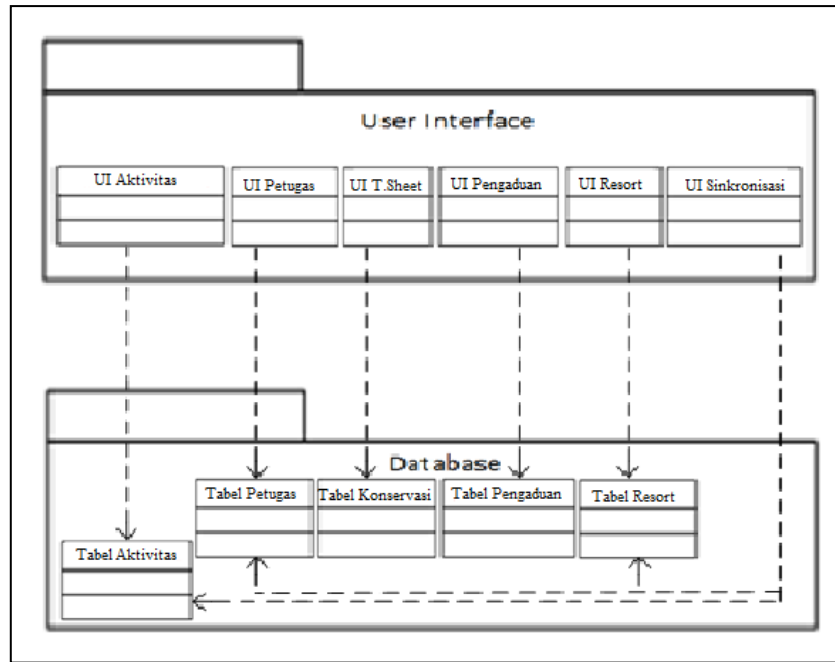


Gambar 4.21 *Object Diagram*

Terlihat pada *object diagram* diatas terdapat beberapa objek yang saling terhubung satu dengan lainnya. Khusus untuk objek *admin*, *operator resort*, *masyarakat* dan *petugas resort* lapangan dimana digeneralisasi menjadi objek *user*. Selanjutnya, masing-masing objek memiliki keterkaitannya masing-masing dengan objek lain seperti objek *aktivitas konservasi* yang berkaitan langsung dengan objek *admin*, *operator resort*, *petugas resort* lapangan.

### 3. Package Diagram

*Package diagram* (diagram paket) adalah sekelompok model elemen-elemen. Paket yang dapat berisi elemen-elemen model yang berlainan, termasuk paket-paket untuk menciptakan atau menggambarkan sifat hirarki. Paket diberi nama yang menggambarkan isinya, oleh karena itu *package* dapat digunakan untuk beberapa keperluan dan kebutuhan.



Gambar 4.22 *Package Diagram*

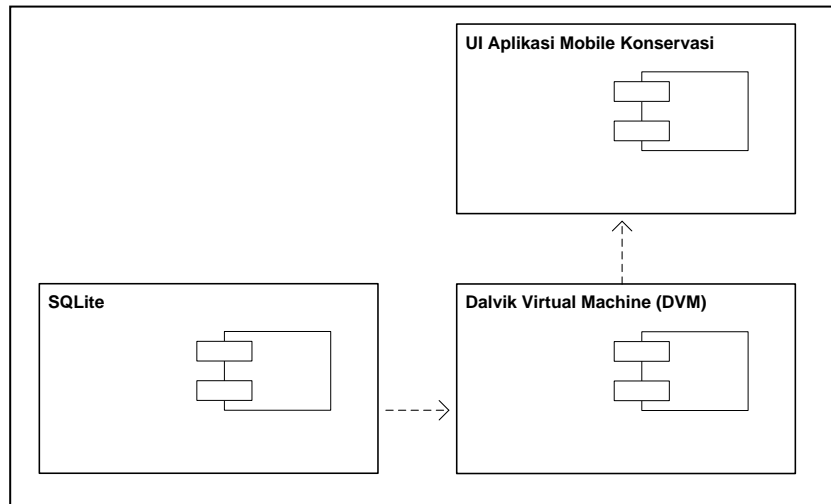
Terlihat jelas pada *package diagram* diatas keterhubungan antara *user interface* (UI) dengan database yang ada, dimana UI aktivitas berhubungan dengan database yang merujuk kepada *table* aktivitas, UI petugas yang juga merujuk pada *table* petugas di *database*, begitupun UI *tally sheet* dan pengaduan merujuk ke *table* konservasi dan pengaduan. Sedangkan UI sinkronisasi tidak hanya merujuk ke satu *table* saja karena berkaitan pula dengan *table* lain seperti *table* aktivitas, *table* petugas dan juga *table* resort pada *database* tersebut.

### 4. Composite Structure Diagram

*Composite Structure Diagram* adalah jenis diagram struktur statis, yang menunjukkan struktur *internal* kelas dan kolaborasi bahwa struktur ini memungkinkan. Namun pada penelitian ini, struktur kelas yang dibuat tidak terlalu kompleks dan hanya melakukan aksi tambah, edit dan hapus sehingga *composite structure diagram* tidak perlu dibuat.

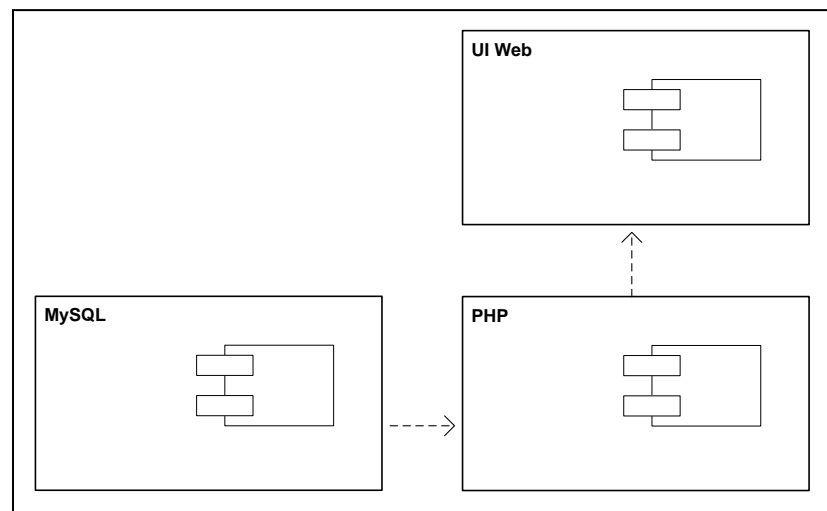
## 5. Component Diagram

*Component diagram* adalah diagram yang digunakan untuk menggambarkan organisasi dan ketergantungan komponen-komponen *software* sistem. *Component diagram* berguna untuk memodelkan komponen objek.



Gambar 4.23 *Component Diagram* Aplikasi Mobile

Terlihat pada *Component Diagram* Aplikasi Mobile diatas menggambarkan organisasi dan ketergantungan komponen-komponen perangkat lunak yang diperlukan untuk menjalankan aplikasi *mobile*. SQLite berguna untuk menyimpan data sementara ketika tidak ada jaringan internet untuk mengirim data. DVM merupakan mesin *virtual* untuk menjalankan aplikasi dan *interface* pada user.



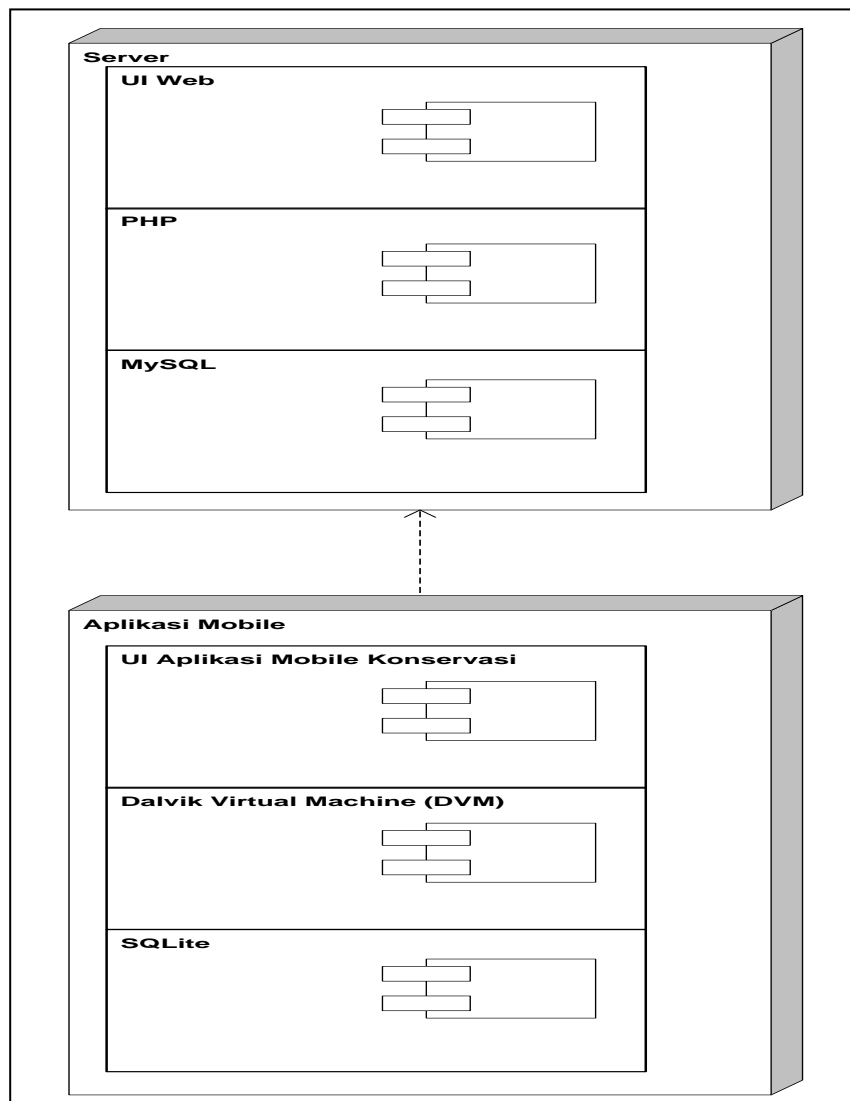
Gambar 4.24 *Component Diagram* Web

Terlihat pada *Component Diagram* Web diatas menggambarkan organisasi dan ketergantungan komponen-komponen perangkat lunak yang diperlukan untuk

menjalankan web. MySQL berguna untuk menyimpan data hasil pengelolaan aktivitas konservasi, maupun manajemen lainnya. Pemrosesan pada web ditangani olehh PHP. *User Interface* web juga bergantung dengan PHP.

#### 6. *Deployment Diagram*

*Deployment Diagram* adalah diagram yang menggambarkan detail bagaimana komponen di-sebar (di-*deploy*) kedalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, *node*, *server* atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal.



Gambar 4.25 *Deployment Diagram*

Pada *Deployment Diagram* diatas terlihat komponen di-sebar (di-*deploy*) kedalam infrastruktur sistem, dimana komponen akan terletak pada server web dan

aplikasi *mobile*. Pemodelan tersebut sangat berguna dan bermanfaat dalam memberikan gambaran sistem *client-server* yang akan dibuat.

#### 4.2.2 Perancangan Database

Perancangan basis data pada penelitian ini melibatkan 15 tabel. Sesuai dengan analisis basis data pada halaman sebelumnya terdapat beberapa keterkaitan dari masing-masing tabel. Pada penjelasan kali ini akan ditampilkan secara detil dari tabel tersebut.

1. Tabel 4.1 Struktur tabel tbl\_bencana

Field	Type	Value	Key
id_bencana	Int	11	*
jenis_aktivitas	Varchar	25	
nama_bencana	Longtext		

2. Tabel 4.2 Struktur tabel tbl\_provinsi

Field	Type	Value	Key
id_prov	Int	11	*
nama_prov	Varchar	10	
keterangan	Longtext		

3. Tabel 4.3 Struktur tabel tb\_aktivitas

Field	Type	Value	Key
id_aktivitas0	Varchar	4	*
id_aktivitas	Varchar	4	**
jenis_aktivitas	Varchar	20	

4. Tabel 4.4 Struktur tabel tb\_batas

Field	Type	Value	Key
id_batas	Int	11	*
id_aktivitas	Int	11	**
id_kawasan	Int	11	**
batas	Varchar	30	

5. Tabel 4.5 Struktur tabel tb\_kawasan

Field	Type	Value	Key
id	Int	11	*
id_kawasan	Int	3	**
id_aktivitas	Int	11	**
kawasan	Varchar	6	
resort	Varchar	30	
koorx	Double		
koory	Double		
akuras	Double		
grid	Vatchar	4	
foto	Longtext		
keterangan	Varchar	100	
waktu	Char	20	
pengirim	Varchar	50	

6. Tabel 4.6 Struktur tabel tb\_kejadian

Field	Type	Value	Key
id_kejadian	Int	11	*
id_aktivitas	Int	11	**
id_kawasan	Int	11	**
kejadiin	Varchar	30	
jenis	Varchar	50	
jumlah	Int	11	
tanda	Varchar	30	
korban	Varchar	100	
kerugian	Varchar	100	

7. Tabel 4.7 Struktur tabel tb\_koordinasi

Field	Type	Value	Key
id_koordinasi	Int	11	*
id_aktivitas	Int	11	**
id_kawasan	Int	11	**
koordinasi	Varchar	20	

8. Tabel 4.8 Struktur tabel tb\_pasangpapan

Field	Type	Value	Key
id_pasangpapan	Int	11	*
id_aktivitas	Int	11	**
id_kawasan	Int	11	**

9. Tabel 4.9 Struktur tabel tb\_pelanggaran

Field	Type	Value	Key
id_pelanggaran	Int	11	*
id_aktivitas	Int	11	**
id_kawasan	Int	11	**
pelanggaran	Varchar	30	
jenis	Varchar	30	
jumlah	Int	5	
pelaku	Varchar	30	
umur	Int	3	
alamat	Varchar	50	

10. Tabel 4.10 Struktur tabel tb\_penyuluhan

Field	Type	Value	Key
id_penyuluhan	Int	11	*
id_aktivitas	Int	11	**
id_kawasan	Int	11	**
penyuluhan	Varchar	100	

11. Tabel 4.11 Struktur tabel tb\_resort

Field	Type	Value	Key
id_resort	Int	11	*
resort	Varchar	100	
keterangan	Varchar	100	
koordinat	Varchar	500	

12. Tabel 4.12 Struktur tabel tb\_sosialisasi

Field	Type	Value	Key
id_koordinasi	Int	11	*
id_aktivitas	Int	11	**
id_kawasan	Int	11	**
sosialisasi	Varchar	100	

13. Tabel 4.13 Struktur tabel tb\_temuan

Field	Type	Value	Key
id_temuan	Int	11	*
id_aktivitas	Int	11	**
id_kawasan	Int	11	**
temuan	Varchar	100	



status	Varchar	15	
jenis	Varchar	40	
tanda	Varchar	20	
tinggi	Double		
diamater	Double		
umur	Int	4	

14. Tabel 4.14 Struktur tabel tb\_tem\_mas

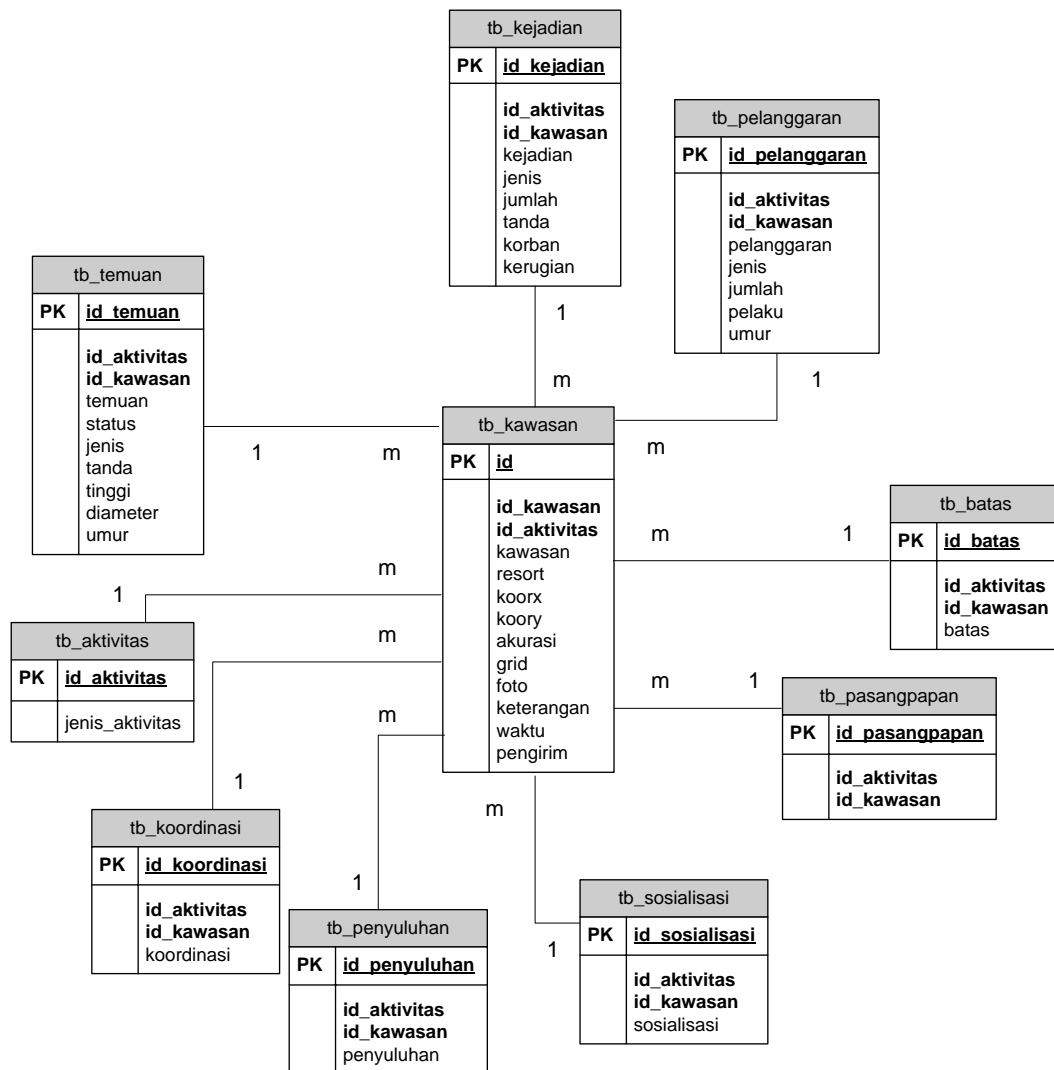
Field	Type	Value	Key
id_tem_mas	Int	11	*
nama	Varchar	50	
alamat	Text		
kontak	Varchar	20	
keterangan	Varchar	100	
koorx	Double		
koory	Double		

15. Tabel 4.15 Struktur tabel tb\_user

Field	Type	Value	Key
id_user	Int	11	*
username	Varchar	20	
password	Varchar	20	

### 4.2.3 Perancangan *Table Relationship*

*Table Relationship* menyatakan hubungan yang terjadi antar table yang ada pada database. Pada penelitian ini terdapat 15 tabel yang beberapa diantaranya memiliki keterkaitan. Hal tersebut dapat dilihat pada *table relationship* berikut :



Gambar 4.26 Perancangan Database

Terlihat pada perancangan basis data diatas terdapat beberapa keterkaitan antar tabel yang gambarkan dengan garis terhubung dengan drajat relasi masing-masing. Adapaun tabel tb\_temuan dan tb\_kawasan contohnya dengan drajat relasi 1 ke m yang berarti kegiatan konservasi temuan terdapat pada beberapa maupun banyak kawasan. Begitu pula dengan tabel lain dengan drajat relasi 1(satu) ke m(banyak) memiliki arti bahwa kegiatan konservasi yang dilakukan terdapat pada banyak kawasan. Adapun model keterhubungan entitas terdapat pada lampiran.

#### 4.2.4 Perancangan User Interface

Perancangan antar muka atau biasa disebut *user interface* merupakan suatu cara yang efektif, dan efisien dalam membangun rancangan perangkat lunak sehingga sesuai dengan apa yang diharapkan dan dibutuhkan oleh pengguna. Pembangunan sistem ini ditujukan untuk mempermudah kegiatan konservasi,

inventarisasi dan visualisasi sebaran flora dan fauna dalam mendukung manajemen berbasis resort. Sistem ini dirancang dengan tampilan ramah pengguna (*user friendly*), dimana menitik beratkan kepada kemudahan pengguna dalam mengoperasikan sistem ini. Berikut ini merupakan rancangan antar muka dari sistem yang akan dibuat.

Perancangan antar muka terdiri dari beberapa form yaitu :

1. Form Awal Aplikasi Mobile Konservasi

Form awal aplikasi *mobile* berisi *username* petugas *resort* lapangan dan *password* dimana telah didaftarkan terlebih dahulu pada web. Terdapat *button* Masuk untuk masuk ke menu utama dan *button* Keluar untuk keluar dari aplikasi *mobile* serta *button* Skip untuk melewati proses autentifikasi.

The diagram shows a rounded rectangular container representing a mobile app screen. At the top, there is a rectangular box with the text "AUTENTIFIKASI". Below this, there are two rows of labels and input fields. The first row has the label "USERNAME" followed by a rectangular input box. The second row has the label "PASSWORD" followed by another rectangular input box. At the bottom of the container, there are three rectangular buttons stacked vertically, each with a small vertical bar on its left side. The buttons are labeled "MASUK", "KELUAR", and "SKIP" from top to bottom.

Gambar 4.27 Form Awal Aplikasi Mobile Konservasi

2. Form Menu Utama Aplikasi Mobile Konservasi

Pada *form* menu utama terdapat beberapa pilihan aktivitas meliputi aktivitas konservasi dimana proses selanjutnya akan merujuk ke kegiatan konservasi yang ada, aktivitas sinkronisasi yang akan merujuk ke proses sinkronisasi data yang telah diperoleh selama melakukan aktivitas konservasi baik yang terkirim maupun tidak

serta aktivitas peta yang berguna untuk *review* titik, rute maupun lokasi aktivitas konservasi yang pernah dilakukan.

The diagram shows a mobile application menu interface. It consists of a rounded rectangular container with a black border. Inside the container, there are five rectangular buttons stacked vertically, each with a black border and bold, uppercase text. The buttons are labeled: 'MENU UTAMA', 'AKTIVITAS KONSERVASI', 'SINKRONISASI', 'PETA', and 'KELUAR'.

Gambar 4.28 *Form Awal Aplikasi Mobile Konservasi*

### 3. *Form Menu Aktivitas Konservasi*

*Form* menu aktivitas konservasi merupakan *form* inti untuk melakukan *input* seluruh data aktivitas konservasi paada aplikasi *mobile*. Terdapat beberapa *input* aktivitas konservasi yang akan dilakukan seperti *input* temuan, pelanggaran, kejadian, batas alam, penyuluhan, sosialisasi, koordinasi dan pemasangan papan peringatan. Adapun dalam rancangan ini akan ditampilkan salah satu contoh *form* aktivitas konservasi yaitu *form input* temuan. Terdapat beberapa *input field* untuk memasukan temuan, status, jenis, tanda, tinggi, diameter serta terdapat pula *button* kirim.

The form is titled "AKTIVITAS KONSERVASI". It contains several input fields and a submit button. The fields are arranged in two columns. The left column has labels: "TEMUAN", "JENIS", "TANDA", "TINGGI", and "DIAMETER". The right column has two dropdown menus (indicated by a checkmark icon) and three text input fields. At the bottom center is a button labeled "KIRIM".

Gambar 4.29 *Form* Aktivitas Konservasi Sebaran Flora dan Fauna

#### 4. *Form* Menu Sinkronisasi

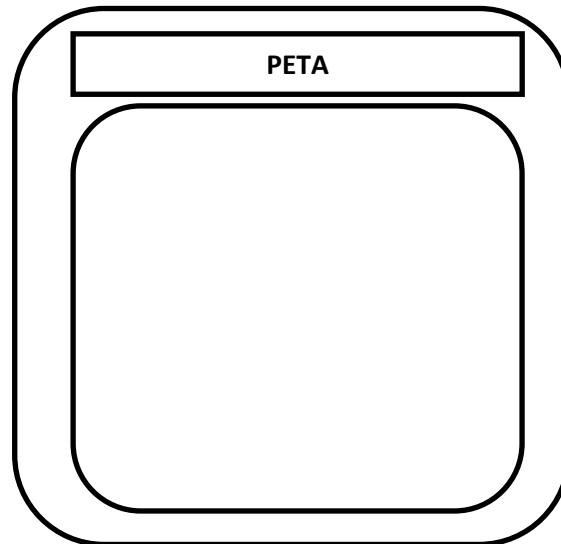
*Form* sinkronisasi merupakan *form* lanjutan dari *form* aktivitas konservasi terkait dengan proses kirim data ke *server*. Dalam hal ini berbagai aktivitas terkait dengan aktivitas konservasi yang tidak terkirim dengan baik akan tersimpan di penyimpanan sementara di *form* sinkronisasi, Terdapat beberapa *button* eksekusi seperti sinkronisasi data dan hapus.

The form is titled "SINKRONISASI". It features a list of three items, each represented by a horizontal bar with a circular icon containing an 'X' on the right side. Below this list are two buttons: "SINKRONISASI" and "HAPUS".

Gambar 4.30 *Form* Aktivitas Konservasi

### 5. *Form Peta*

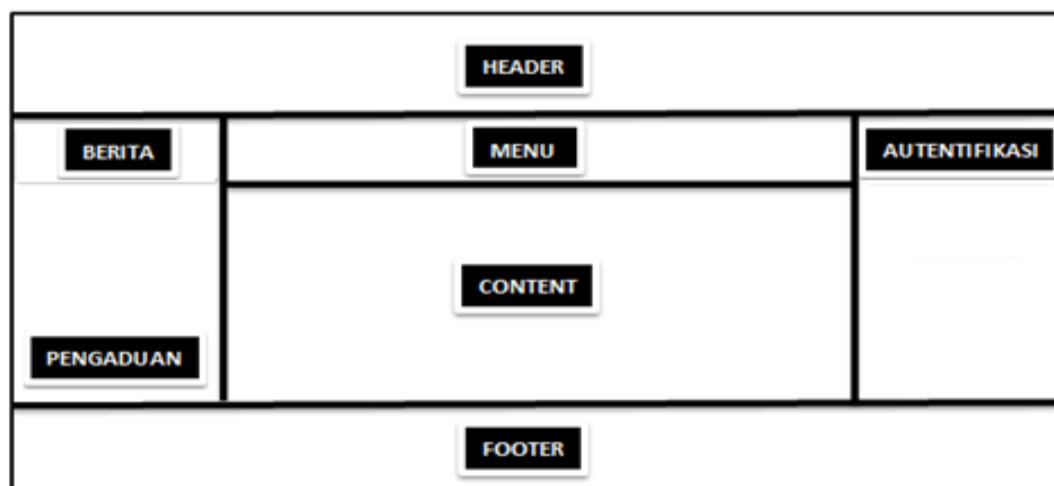
*Form Peta* merupakan *form* untuk menampilkan peta dimana data yang ditampilkan pada peta berasal dari *form input* data pada menu aktivitas konservasi sebelumnya. Adapun rancangan *form* peta adalah sebagai berikut :



Gambar 4.31 *Form Peta*

### 6. *Form Utama Web*

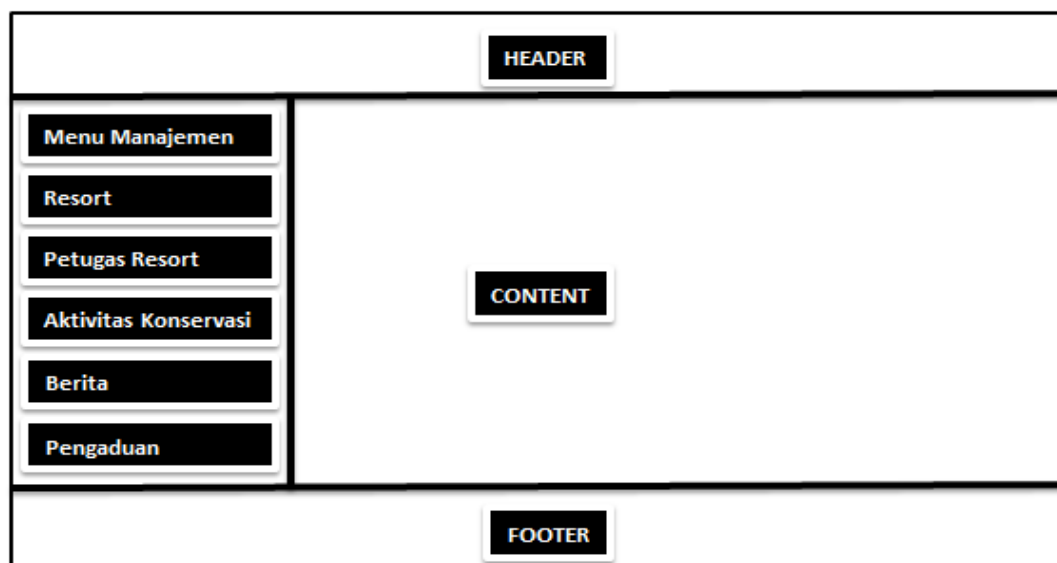
*Form* utama web berisi informasi umum terkait dengan aktivitas konservasi yang dilakukan oleh Balai Konservasi ditingkat *resort*, termasuk menampilkan informasi, peta *resort*, hasil aktivitas konservasi, galeri foto, serta layanan pengaduan oleh masyarakat. Terdapat pula *field* yang digunakan untuk akses masuk kedalam sesi *operator resort* maupun admin web.



Gambar 4.32 *Form Utama Web*

## 7. Form Utama Admin

*Form* utama admin merupakan *form* manajemen terhadap sistem yang dapat mengatur hampir seluruh manajemen yang ada. Pada *form* utama admin terdapat lima pilihan manajemen meliputi manajemen aktivitas , manajemen *resort*, manajemen petugas, manajemen aktivitas konservasi, manajemen pengaduan dan menampilkan peta, rute serta statistik. Masing-masing manajemen terlibat langsung dengan seluruh kegiatan aktivitas konservasi pada penelitian ini dimulai dari integrasi aktivitas konservasi yang dilakukan pada aplikasi *mobile* maupun pada web oleh *operator resort*. Dalam manajemen ini terdapat pilihan untuk memperbaiki, menambah, maupun menghapus aktivitas konservasi yang telah dilakukan. Selanjutnya, merujuk ke manajemen *resort* yang berkaitan dengan tata ruang *resort* terdapat pilihan untuk memperbaiki, menambah maupun menghapus *resort* yang ada. Begitu juga dengan manajemen petugas *resort*, manajemen aktivitas dan manajemen pengaduan serta menampilkan peta, rute dan statistic yang dapat dikelola oleh admin melalui *form* ini.



Gambar 4.33 *Form* Utama Admin

## 8. Form Utama Operator

*Form* utama operator merupakan *form* manajemen sistem oleh operator yang dapat mengatur sebagian manajemen yang ada. Pada *form* utama operator terdapat tiga pilihan utama meliputi manajemen petugas *resort*, manajemen aktivitas konservasi dan penampilan peta. Dalam manajemen aktivitas konservasi

di *form operator resort* terdapat pilihan untuk memperbaiki, menambah, maupun menghapus aktivitas konservasi. Selanjutnya, merujuk ke manajemen petugas *resort* yang ada terdapat pilihan untuk memperbaiki, menambah maupun menghapus petugas *resort* yang ada di masing-masing *resort*. *Form operator resort* membantu apabila aktivitas konservasi yang dilakukan oleh petugas *resort* lapangan terkendala masalah jaringan maupun cuaca sehingga proses aktivitas konservasi tetap bisa dilaksanakan. Selanjutnya, terdapat menu penampilan peta *resort* yang berisi wilayah *resort* dan titik pengumpulan data aktivitas konservasi.

HEADER
Menu Manajemen
CONTENT
FOOTER

Gambar 4.34 *Form Utama Operator*



## BAB V

### HASIL DAN PEMBAHASAN

Sesuai dengan model pengembangan sistem yang digunakan yaitu sequensial linier atau model *waterfall* (Sommerville, 2001). Adapun tahapan-tahapan pengembangan sistem yang telah dilakukan pada bab sebelumnya mengenai (1) *Requirements analysis and Definition* (Definisi dan Analisis Kebutuhan), (2) *System and Software Design* (Desain Sistem dan Perangkat Lunak), pada bab ini akan dilakukan pembahasan selanjutnya yaitu (3) *Implementation and Unit Testing* (Implementasi dan Pengujian Unit), (4) *Integration and System Testing* (Integrasi dan Pengujian Sistem), dan (5) *Operation and Maintenance* (Operasi dan Pemeliharaan).

#### 5.1 Implementasi Sistem

##### 5.1.1 Penulisan Kode Program (*Coding*)

Pada implementasi sistem, kegiatan diawali dengan penulisan kode program (*coding*). Penulisan kode program (*coding*) merupakan tahapan hasil analisa dan perancangan sistem awal hingga menjadi suatu sistem yang utuh dan terintegrasi. Penulisan program yang digunakan pada sistem ini menggunakan bahasa *Java* dengan tool *IDE Eclipse Indigo (Platform Version 3.7)* yang terintegrasi dengan *Android Software Development Kit (Android SDK)* dan *Android Development Tools (ADT)* serta bahasa pemrograman web *PHP (Hypertext Preprocessor)*



Gambar 5.1 IDE Eclipse Indigo

Penulisan kode di IDE *eclipse* terbagi menjadi 2 (dua) bagian yaitu pembuatan kelas menggunakan bahasa *Java* dan pembuatan tampilan (*layout*) sistem itu menggunakan bahasa *XML*. Sehingga dalam suatu *project* terdapat dua *ekstensi* kode program yaitu *.java* dan *.xml*. Selanjutnya, penulisan kode PHP dilakukan pada *Notepad++* dengan *browser Mozilla Firefox* sebagai *compiler* hasilnya.

## 1. Pembuatan *Login Aplikasi*

### 1) Kelas *Password*

Kelas *Password* adalah kelas yang pertama dipanggil saat aplikasi dijalankan. Kelas ini berisikan kode yang digunakan untuk menciptakan autentifikasi untuk masuk kedalam aplikasi. Kemudian jika proses autentifikasi selesai maka akan dipanggil kelas Menu yaitu kelas untuk menampilkan beranda menu. Proses pemanggilan kelas baru dilakukan dengan cara *intent* yaitu menghubungkan antar kelas pada *project* sistem. Adapun kode programnya sebagai berikut,

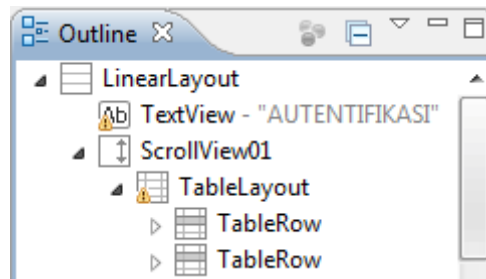
```
Intent Enter=new Intent (this,
menu.class);

startActivity(Enter);
```

Maksud dari kode program diatas adalah melakukan pemanggilan kelas dari kelas Password ke kelas Menu.

## 2) *Layout Password*

Untuk pembuatan tampilan atau *layout* dituliskan dengan format *XML*. *Layout Password* berisikan kode untuk menampilkan *field input data (EditText)*, dan button submit (*Button*).



Gambar 5.2 Struktur XML Tampilan Password

## 2. Pembuatan Menu

### 1) Kelas Menu

Kelas menu adalah kelas yang dipanggil setelah kelas *Password*. Kelas ini menampilkan menu pilihan yang dapat digunakan oleh *user*. Seperti yang telah dirancang sebelumnya bahwa sistem ini dibuat dengan 4 (empat) pilihan menu. Menu-menu pilihan tersebut dipanggil dengan *onClick*. Karena jumlah menu lebih dari satu maka menu-menu tersebut dipanggil dengan fungsi *switch*. Berikut kode programnya :

```

@Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        switch (v.getId()) {

            case R.id.aktivitas:
                Intent Enter=new Intent (this, aktivitas.class);
                startActivity(Enter);
                break;

            case R.id.peta:
                Intent Enter1=new Intent (this, peta.class);
                startActivity(Enter1);
                break;

            case R.id.sinkronisasi:
                Intent Enter2=new Intent (this,
sinkronisasi.class);
                startActivity(Enter2);
                break;

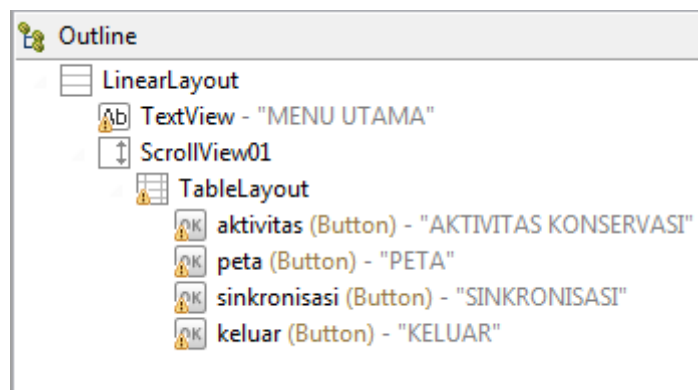
            case R.id.keluar:
                finish();
                break;
        }
    }

```

Sperti yang telah dijelaskan bahwa masing-masing menu yang dibuat diberi **case R.id.** yang merujuk pada pemanggilan kelas khusus yang berbeda, dilanjutkan dengan pembuatan objek baru untuk memanggil kelas tersebut dengan Intent.

## 2) Layout Menu

Adapun struktur *xml* dari *layout* menu sebagai berikut :



Gambar 5.3 Struktur XML Tampilan Beranda Menu

Terlihat di struktur *xml* yang menjadi layout induk adalah *LinearLayout* dan *ScrolView*. Untuk *header* nama aplikasi terletak di *LinearLayout* yang berupa *TextView*. Sedangkan untuk menu-menu pilihan yang berupa *button* (tombol) terletak di *TableLayout* yang disusun secara *center horizontal*.

### 3. Pembuatan Menu Aktivitas Konservasi.

Menu aktivitas konservasi terdiri dari beberapa kelas. Pada menu ini terdapat kelas Aktivitas yang terdiri dari 2 (dua) menu utama yaitu menu Patroli dan menu Penyuluhan dimana masing-masing menu terdapat 4 (empat) submenu dengan kelasnya masing-masing. Pada menu Patroli terdapat 4 (empat) kelas yaitu kelas Temuan, kelas Pelanggaran, kelas Kejadian dan kelas Batas. Pada menu Penyuluhan juga terdapat 4 (empat) kelas yaitu kelas Penyuluhan, kelas Sosialisasi, kelas Koordinasi dan kelas Pasangpapan. Berikut merupakan penjelasan dari masing-masing kelas tersebut :

#### 1) Kelas Temuan

Kelas Temuan adalah kelas yang digunakan sebagai *input* data temuan flora dan fauna pada aktivitas konservasi. Kelas ini berfungsi sebagai tempat pengumpulan data lengkap terkait dengan temuan flora dan fauna, terdiri dari *field* daftar flora dan fauna dilindungi, *auto-generated* koordinat, pengambilan foto dan isian data yang nantinya akan disimpan dan ditampilkan di peta (*GoogleMaps*) serta dikirim ke web. Adapun potongan kode program dari kelas Temuan sebagai berikut :

```
String sql="INSERT INTO tb_temuan
(id_aktivitas,temuan,status,jenis,tanda,tinggi,diamet
er,umur,id_kawasan)VALUES
('"+id_patroli+"','"+spin.getSelectedItem().toString(
)+"','"+spin1.getSelectedItem().toString()+"','"+edit
.getText().toString()+"','"+spin2.getSelectedItem().t
oString()+"','"+edit1.getText().toString()+"','"+edit
2.getText().toString()+"','"+edit3.getText().toString
()+"','"+id_kawasan+"'");

db.masukinData(sql);
id_temuan=db.ambilidtemuan();
kirim_data();
clearform();
```

Potongan kode diatas menunjukkan proses *insert* data ke tabel temuan di *database* berdasarkan *field* yang telah disesuaikan. Dilanjutkan dengan pengambilan id temuan secara otomatis, melakukan pengiriman data kembali setelah disimpan dan mengosongkan *form* seperti semula.

## 2) Kelas Pelanggaran

Kelas Pelanggaran adalah kelas yang digunakan sebagai *input* data kegiatan pelanggaran dikawasan maupun diluar kawasan konservasi. Kelas ini berfungsi sebagai tempat pengumpulan data lengkap terkait dengan pelanggaran konservasi alam, terdiri dari *field input* data pelanggaran, *auto-generated* koordinat, pengambilan foto dan isian data lainnya yang nantinya akan simpan dan ditampilkan di peta (*GoogleMaps*) serta dikirim ke web. Adapun potongan kode program dari kelas Pelanggaran sebagai berikut :

```
String sql="INSERT INTO tb_pelanggaran
(id_aktivitas,pelanggaran,jenis,jumlah,pelaku,umur,al
amat,id_kawasan)VALUES
('"+id_patroli+"','"+spin.getSelectedItem().toString(
)+"','"+edit.getText().toString()+"','"+edit1.getText
().toString()+"','"+edit7.getText().toString()+"','"+
edit8.getText().toString()+"','"+edit9.getText().toSt
ring()+"','"+id_kawasan+"'");

db.masukinData(sql);

id_pelanggaran=db.ambilidpelanggaran();
kirim_data();
clearform();
```

Potongan kode diatas menunjukan proses *insert* data ke tabel pelanggaran di *database* berdasarkan *field* yang telah disesuaikan. Dilanjutkan dengan pengambilan id temuan secara otomatis, melakukan pengiriman data kembali setelah disimpan dan mengosongkan *form* seperti semula.

### 3) Kelas Kejadian

Kelas Kejadian adalah kelas yang digunakan sebagai *input* data kejadian terfokus pada konflik satwa dan titik api/kebakaran dikawasan maupun diluar kawasan konservasi. Kelas ini berfungsi sebagai tempat pengumpulan data lengkap terkait dengan kejadian tersebut, terdiri dari *field input* data kejadian, *auto-generated* koordinat, pengambilan foto dan isian data lainnya yang nantinya akan simpan dan ditampilkan di peta (*GoogleMaps*) serta dikirim ke web. Adapun potongan kode program dari kelas Kejadian sebagai berikut :

```
String sql="INSERT INTO tb_kejadian
(id_aktivitas,kejadian,jenis,jumlah,tanda,korban,keru
gian,id_kawasan)VALUES
('"+id_patroli+"','"+spi.getSelectedItemAt().toString()
+"','"+konflik.getText().toString()+"','"+api.getText
().toString()+"','"+spi3.getSelectedItemAt().toString()
+"','"+korban.getText().toString()+"','"+kerugian.get
Text().toString()+"','"+id_kawasan+"')";
    db.masukinData(sql);

    id_kejadian=db.ambilidkejadian();
    kirim_data();
    clearform();
```

Potongan kode diatas menunjukan proses *insert* data ke tabel kejadian di *database* berdasarkan *field* yang telah disesuaikan. Dilanjutkan dengan pengambilan id temuan secara otomatis, melakukan pengiriman data kembali setelah disimpan dan mengosongkan *form* seperti semula.

#### 4) Kelas Batas

Kelas Kejadian adalah kelas yang digunakan sebagai *input* data kejadian terfokus pada konflik satwa dan titik api/kebakaran dikawasan maupun diluar kawasan konservasi. Kelas ini berfungsi sebagai tempat pengumpulan data lengkap terkait dengan kejadian tersebut, terdiri dari *field input* data kejadian, *auto-generated* koordinat, pengambilan foto dan isian data lainnya yang nantinya akan disimpan dan ditampilkan di peta (*GoogleMaps*) serta dikirim ke web. Adapun potongan kode program dari kelas Kejadian sebagai berikut :

```
String sql="INSERT INTO tb_batas
(id_aktivitas,batas,id_kawasan)VALUES
('"+id_patroli+"','"+spin.getSelectedItem().toString(
)+"','"+id_kawasan+"')";

    id_batas=db.ambilidbatas();
    db.masukinData(sql);
    kirim_data();
    clearform();
```

Potongan kode diatas menunjukan proses *insert* data ke tabel batas di *database* berdasarkan *field* yang telah disesuaikan. Dilanjutkan dengan pengambilan id temuan secara otomatis, melakukan pengiriman data kembali setelah disimpan dan mengosongkan *form* seperti semula.

#### 5) Kelas Penyuluhan

Kelas Penyuluhan adalah kelas yang digunakan sebagai *input* data penyuluhan baik perorangan maupun kelompok terhadap masyarakat, institusi maupun lembaga dikawasan maupun diluar kawasan konservasi. Kelas ini berfungsi sebagai tempat pengumpulan data lengkap terkait dengan penyuluhan tersebut, terdiri dari *field input* data penyuluhan, *auto-generated* koordinat, pengambilan foto dan isian data lainnya yang nantinya akan disimpan dan



ditampilkan di peta (*GoogleMaps*) serta dikirim ke web. Adapun potongan kode program dari kelas *Penyuluhan* sebagai berikut :

```
String sql="INSERT INTO tb_penyuluhan
(id_aktivitas,penyuluhan,id_kawasan)VALUES
('"+id_penyuluhan0+"','"+spi.getSelectedItem().toString()+"','"+id_kawasan+"')";

db.masukinData(sql);

id_penyuluhan=db.ambilidpenyuluhan();
kirim_data();
clearform();
```

Potongan kode diatas menunjukkan proses *insert* data ke tabel *penyuluhan* di *database* berdasarkan *field* yang telah disesuaikan. Dilanjutkan dengan pengambilan id temuan secara otomatis, melakukan pengiriman data kembali setelah disimpan dan mengosongkan *form* seperti semula.

#### 6) Kelas Sosialisasi

Kelas Sosialisasi adalah kelas yang digunakan sebagai *input* data sosialisasi baik perorangan maupun kelompok terhadap masyarakat, institusi maupun lembaga dikawasan maupun diluar kawasan konservasi. Kelas ini berfungsi sebagai tempat pengumpulan data lengkap terkait dengan sosialisasi tersebut, terdiri dari *field input* data Sosialisasi, *auto-generated* koordinat, pengambilan foto dan isian data lainnya yang nantinya akan simpan dan ditampilkan di peta (*GoogleMaps*) serta dikirim ke web. Adapun potongan kode program dari kelas Sosialisasi sebagai berikut :

```
String sql="INSERT INTO tb_sosialisasi
(id_aktivitas,sosialisasi,id_kawasan)VALUES
('"+id_penyuluhan0+"','"+spi.getSelectedItem().toString()+"','"+id_kawasan+"')";

db.masukinData(sql);

id_sosialisasi=db.ambilidsosialisasi();
kirim_data();
clearform();
```

Potongan kode diatas menunjukkan proses *insert* data ke tabel *sosialisasi* di

*database* berdasarkan *field* yang telah disesuaikan. Dilanjutkan dengan pengambilan id temuan secara otomatis, melakukan pengiriman data kembali setelah disimpan dan mengosongkan *form* seperti semula.

#### 7) Kelas Koordinasi

Kelas Koordinasi adalah kelas yang digunakan sebagai *input* data koordinasi baik perorangan maupun kelompok terhadap masyarakat, institusi maupun lembaga dikawasan maupun diluar kawasan konservasi. Kelas ini berfungsi sebagai tempat pengumpulan data lengkap terkait dengan koordinasi tersebut, terdiri dari *field input* data Koordinasi, *auto-generated* koordinat, pengambilan foto dan isian data lainnya yang nantinya akan simpan dan ditampilkan di peta (*GoogleMaps*) serta dikirim ke web. Adapun potongan kode program dari kelas Koordinasi sebagai berikut :

```
String sql="INSERT INTO tb_koordinasi
(id_aktivitas,koordinasi,id_kawasan)VALUES
('"+id_penyuluhan0+"','"+spi.getSelectedItem().toString()+"','"+id_kawasan+"'");

db.masukinData(sql);

id_koordinasi=db.ambilidkoordinasi();
kirim_data();
clearform();
```

Potongan kode diatas menunjukan proses *insert* data ke tabel koordinasi di *database* berdasarkan *field* yang telah disesuaikan. Dilanjutkan dengan pengambilan id temuan secara otomatis, melakukan pengiriman data kembali setelah disimpan dan mengosongkan *form* seperti semula.

#### 8) Kelas Pasangpapan

Kelas Pasangpapan adalah kelas yang digunakan sebagai *input* data pemasangan papan peringatan dikawasan maupun diluar kawasan konservasi. Kelas ini berfungsi sebagai tempat pengumpulan data lengkap terkait dengan pemasangan papan tersebut, terdiri dari *field input* data Pasangpapan, *auto-generated* koordinat, pengambilan foto dan isian data lainnya yang nantinya akan simpan dan ditampilkan di peta (*GoogleMaps*) serta dikirim ke web. Adapun potongan kode program dari kelas Pasangpapan sebagai berikut :

```
String sql="INSERT INTO tb_pasangpapan
(id_aktivitas,id_kawasan)VALUES
('"+id_penyuluhan0+"','"+id_kawasan+"')";

db.masukinData(sql);

id_pasangpapan=db.ambilidpasangpapan();
kirim_data();
clearform();
```

Potongan kode diatas menunjukan proses *insert* data ke tabel pasang papan di *database* berdasarkan *field* yang telah disesuaikan. Dilanjutkan dengan pengambilan id temuan secara otomatis, melakukan pengiriman data kembali setelah disimpan dan mengosongkan *form* seperti semula.

#### 4. Pembuatan Menu Sinkronisasi.

##### 1) Kelas Sinkronisasi

Kelas Sinkronisasi didesain untuk mengantisipasi kemungkinan permasalahan teknis yang terjadi di pada proses pengiriman data ke web seperti tidak ada sinyal pengiriman data (*out of range signal*). Kelas ini menjamin bahwa setiap aktivitas yang dilakukan oleh masing-masing kelas *input* data akan *direcord*

kedalam sebuah daftar (*ListView*) sehingga semua data hasil *input* data aktivitas konservasi dapat dimanipulasi untuk dihapus maupun di sinkronisasi kembali.

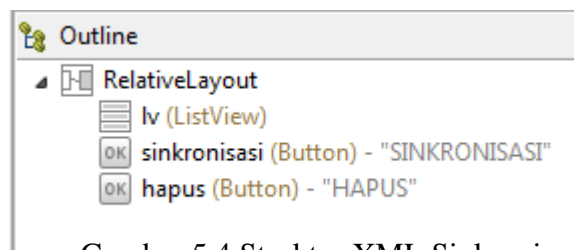
Berikut potongan program dari kelas sinkronisasi :

```
private void kirim_data2() {  
    String  
    url="http://webgis2013.pusku.com/admin/sinkronisasi.  
    php";  
  
    HttpClient klien=new DefaultHttpClient();  
    HttpPost kirim=new HttpPost(url);  
  
    ambildatatemuan();  
    ambildatapelanggaran();  
    ambildatakejadian();  
    ambildatabatas();  
    ambildatapenyuluhan();  
    ambildatasosialisasi();  
    ambildatakoordinasi();  
    ambildatapapan();  
    ambildatakawasan();  
}
```

Potongan kode diatas menjelaskan pengiriman data ke web menggunakan fungsi `HttpClient` dan `HttpPost` yang mengambil data dari aktivitas konservasi yang dilaksanakan.

## 2) Layout Sinkronisasi.

Untuk layout Sinkronisasi menggunakan *RelativeLayout* dan diikuti dengan *ListView* pada file *xml*. Hal ini berfungsi untuk melakukan penempatan yang dinamis serta menghasilkan daftar data *vertical* yang masuk dengan tambahan *multiple choice checkbox*. Adapun struktur penyusunan *xml* sebagai berikut :

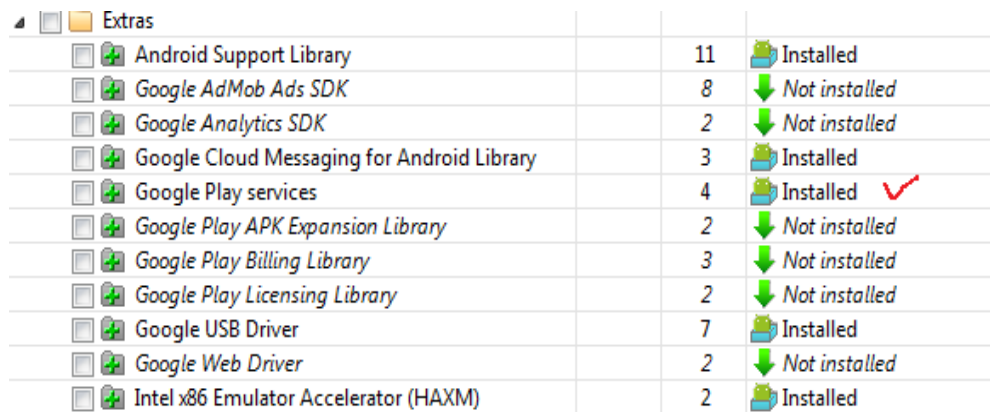


Gambar 5.4 Struktur XML Sinkronisasi

## 5. Pembuatan Menu Peta

Untuk pembuatan menu Peta terdapat beberapa tahapan konfigurasi yang harus dilakukan. Penampilan peta pada menu ini menggunakan *Google Maps API* v.2 dimana merupakan versi terbaru penampilan peta memperbaiki v.1 yang telah tidak lagi digunakan (*deprecated*). Berbeda dengan penampilan peta pada versi sebelumnya, pada API v.2 perlu dilakukan beberapa tahapan. Adapun tahapan tersebut adalah sebagai berikut :

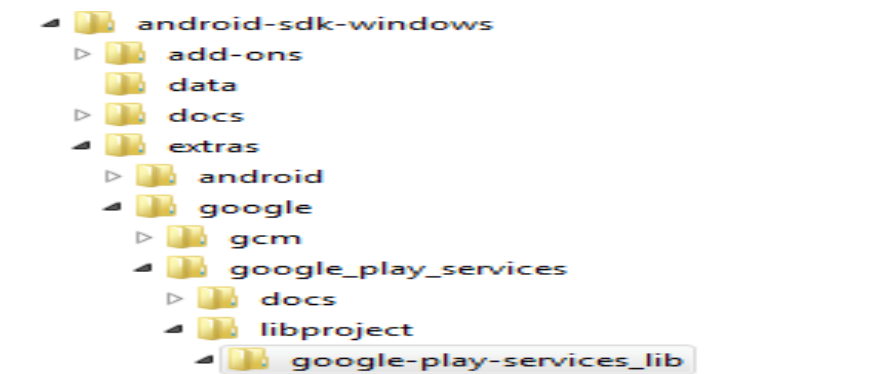
- 1) Melakukan instalasi Google Play Services di SDK Manager.



Extras			
<input type="checkbox"/>	Android Support Library	11	Installed
<input type="checkbox"/>	Google AdMob Ads SDK	8	Not installed
<input type="checkbox"/>	Google Analytics SDK	2	Not installed
<input type="checkbox"/>	Google Cloud Messaging for Android Library	3	Installed
<input type="checkbox"/>	Google Play services	4	Installed ✓
<input type="checkbox"/>	Google Play APK Expansion Library	2	Not installed
<input type="checkbox"/>	Google Play Billing Library	3	Not installed
<input type="checkbox"/>	Google Play Licensing Library	2	Not installed
<input type="checkbox"/>	Google USB Driver	7	Installed
<input type="checkbox"/>	Google Web Driver	2	Not installed
<input type="checkbox"/>	Intel x86 Emulator Accelerator (HAXM)	2	Installed

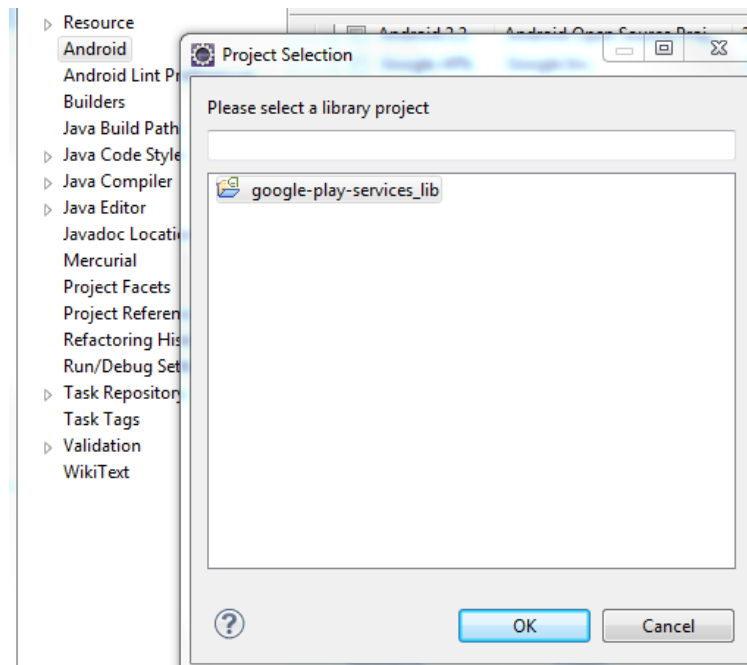
Gambar 5.5 Instalasi *Google Play Services*

- 2) Import library yang diperlukan dimana *google* sudah menyiapkan *library* tersendiri untuk Maps. Klik Menu *File Import > General > Existing Project into Workspace > Browse* masuk ke direktori Android SDK.



Gambar 5.6 *Import Library Google Play Services*

3) Selanjutnya, menambahkan *library Google Play* yang sebelumnya untuk diimport ke project. Klik Kanan Project > Properties > Masuk ke bagian android > Add.



Gambar 5.7 Import Library Google Play Services ke Project

4) Mendapatkan *API Key* untuk Google Maps. API key berfungsi sebagai tanda khusus (*fingerprint signature*) untuk penggunaan Aplikasi Android maupun web. Masuk ke URL <https://code.google.com/apis/console/> dan pastikan bahwa Google Maps Android API v.2 dalam keadaan ON.

	Google Maps Android API v2		
	Google Maps API v2		Courtesy limit: 25,000 requests/day • <a href="#">Pricing</a>

Gambar 5.8 Menu Service Google Maps API

5) Mendapatkan *SHA1 certificate fingerprint*. Masuk ke *command prompt* dan *debug.keystore*, pada sistem operasi *windows*, secara *default* posisi *debug.keystore* berada di folder *.android*. Tuliskan kode sebagai berikut :

```
keytool -list -alias androiddebugkey -keystore
debug.keystore -storepass android -keypass
android -v
```

6) Ambil kunci SHA1 yang dan kembali ke browser. Masukkan kunci *SHA1* dan ditambahi nama *package* aplikasi. Adapun contoh pengambilan kunci *SHA1* adalah sebagai berikut :

```
26:08:95:CF:15:FF:88:6B:27:3C:FC:6F:6D:A5:45:FF:6  
1:3C:DD:91;com.aktivitas
```

*com.aktivitas* adalah nama package aplikasi. Adapun hasil akses API sebagai berikut :

Key for Android apps (with certificates)	
API key:	AIzaSyA-iC72eNEI2_fPRoCTrjukaKEifOoH840
Android apps:	26:08:95:CF:15:FF:88:6B:27:3C:FC:6F:6D:A5:45:FF:61:3C:DD:91;com.aktivitas
Activated on:	Jul 2, 2013 6:35 PM
Activated by:	ejoimandeka@gmail.com – you
Key for browser apps (with referers)	
API key:	AIzaSyAMDECb3aSu8S4MakN4yZI--Mr2-1uSXXY
Referers:	Any referer allowed
Activated on:	Apr 14, 2013 8:58 PM
Activated by:	ejoimandeka@gmail.com – you

Gambar 5.9 API Key Android dan Browser

Api Key yang sudah didapat diletakan pada *AndroidManifest.xml* untuk menjalankan fitur peta di *mobile Android* dan untuk di web dideklarasikan diawal penulisan kode PHP, kendati demikian penampilan peta di emulator eclipse masih belum mendukung. Peta dapat ditampilkan langsung pada aplikasi mobile yang telah terinstal.

a. Kelas Peta

Kelas ini merupakan kelas utama dalam menampilkan peta dimana data koordinat berasal dari *form input* data pada menu aktivitas konservasi sebelumnya. Pada kelas ini, peta menampilkan *marker* yang menyatakan titik

dan informasi dimana data *input* dimasukan. Penampilan *marker* berfungsi sebagai gambaran petugas dalam pengambilan data di lapangan. Adapun potongan kode kelas Peta sebagai berikut :

```
public void carilokasi()
{
    LocationManager lok
    =(LocationManager) getSystemService(LOCATION_SERVICE)
    ;

    String locationProvider =
    LocationManager.GPS_PROVIDER;
    lok.requestLocationUpdates(locationProvider,
    1000000, 10, locLis);
    }

    public void onLocationChanged(Location lokasi) {
    if(lokasi !=null){
    double latitude=lokasi.getLatitude();
    double longitude=lokasi.getLongitude();
    Log.d("Lokasi2", latitude+"-"+longitude);
    LatLng LatLng2 = new LatLng (latitude, longitude);
    googleMap.addMarker(new MarkerOptions()
    .position(LatLng2).title("Lokasi")
    .snippet("Saya disini !")

    .icon(BitmapDescriptorFactory.defaultMarker(BitmapDe
    scriptorFactory.HUE_RED)));
    }
```

Potongan kode diatas menjelaskan pencarian lokasi menggunakan *LocationManager* dengan *LocationManager.GPS\_PROVIDER* yang menunjukan pengambilan koordinat berdasarkan GPS dengan *requestLocationUpdates* (*locationProvider, 1000000, 10, locLis*) yang menjelaskan durasi *update* data. Selanjutnya pemberian informasi koordinat yang disajikan dengan *icon default marker* berwarna merah.

#### b. *Layout* Peta

*Layout* peta hanya berisi *fragment* dimana berfungsi sebagai wadah untuk penampilan peta.



c. *DBHelper*

*DBHelper* merupakan kelas bantuan dimana ditugaskan sebagai tempat peletakan berbagai *method* yang digunakan oleh berbagai kelas yang ada. Sehingga *method* yang ada dimasing-masing kelas hanya merupakan pemanggilan dari *method* yang sudah ditanamkan pada kelas *DBHelper* ini. Adapun potongan kode pada kelas ini sebagai berikut :

```
public void createDataBase() throws IOException{
    boolean dbExist = checkDataBase();
    if(dbExist){
    }else{
        this.getReadableDatabase();
        try {
            copyDataBase();
        } catch (IOException e) {
            throw new Error("Error copying
database");
        } } }

public Cursor ambilkoordinat(){

    String selectQuery = "SELECT
a.koorx,a.koory,b.jenis_aktivitas FROM tb_kawasan a
INNER JOIN tb_aktivitas b ON
a.id_aktivitas=b.id_aktivitas";
    Cursor cursor = myDataBase.rawQuery(selectQuery,
null);
    if(cursor != null){
        return cursor;
    } else {
        return null;    } }
```

Potongan kode diatas menjelaskan pembuatan *database* dengan fungsi `createDataBase()` serta melakukan pemeriksaan *database* dilanjutkan dengan aksi `getReadableDatabase` apabila *database* sudah ada dan *error copying* apabila terjadi masalah. Sedangkan potongan kode selanjutnya `selectQuery` adalah untuk menampilkan informasi dan koordinat dari tabel kawasan.

## 6. Pembuatan Menu Keluar Aplikasi

Menu keluar aplikasi tidak memiliki kelas ataupun *layout*, dikarenakan menu keluar aplikasi merupakan bentuk *dialog*. Jika tombol menu keluar aplikasi ditekan maka akan keluar kembali pada menu Password dan dilanjutkan dengan keluar dari aplikasi. Berikut potongan kode programnya :

```
case R.id.keluar:  
    finish();  
    break;
```

Potongan kode sederhana diatas menggunakan fungsi `finish()` pada `case R.id.keluar` mengindikasikan proses keluar dari program.

## 7. Pembuatan Website (WebGIS 2013)

WebGIS 2013 merupakan sebuah *website* yang menampilkan hasil penginputan data dari aplikasi *android* yang telah dibuat sebelumnya. Dalam web ini tidak hanya menampilkan data hasil dari aplikasi *android* tetapi juga manajemen pengelolaan, penampilan peta, serta serta statistik kegiatan. Dalam web yang dibuat terdapat beberapa antarmuka yang dirancang meliputi : antarmuka pengguna umum (*user*), antarmuka admin (*administrator*) serta antarmuka operator. Berikut merupakan penjelasan masing-masing antarmuka yang telah dibuat :

### 1) Antarmuka Admin (administrator)

Pada antarmuka admin terdapat 10 (sepuluh) menu utama yang terdiri dari: beranda, aktivitas, petugas, resort, *tallysheet*, peta, rute, statistik, pengaduan dan *logout*. Masing-masing menu memiliki fungsi masing-masing yang terkait baik untuk menampilkan informasi maupun manajemen data. Berikut penjelasan masing-masing menu tersebut :

a. Menu Beranda

Menu ini berfungsi sebagai antarmuka pertama yang menunjukkan halaman awal admin. Pada menu ini hanya menampilkan judul dari web yang dirancang dan logo dari aktivitas konservasi yang dilakukan yaitu logo PHKA (Perlindungan Hutan dan Konservasi Alam). Adapun potongan kode dari menu beranda (*home.php*) adalah sebagai berikut :

```
<?php
include "../img/index.php";
$echo = base64_decode($acak);
?>

<table width="1000" border="0" align="center"
cellpadding="0" cellspacing="0">

    <tr>

        <p align="center"><b>WEBGIS SEBARAN FLORA, FAUNA &
AKTIVITAS KONSERVASI PROVINSI BENGKULU</b> </p>

        <p align="center" ><b>BERBASIS RBM (<i>RESORT BASED
MANAGEMENT</i></b></p>

        <td align="left" valign="top" style="padding-
top:0px;" class="text_left"><div align="left">

            <p align="center"><br />

            </p>

            <p align="center">&nbsp;</p>

        </div></td>

    </tr>

</table>
```

Potongan kode `include "../img/index.php"` menjelaskan pemanggilan kelas `index.php` yang akan digunakan pada menu beranda. Dilanjutkan dengan kostumasi tampilan pada halaman tersebut.

## b. Menu Aktivitas Konservasi

Menu ini berfungsi sebagai tempat penambahan, penghapusan maupun pengubahan aktivitas konservasi yang dilakukan pada penelitian ini, menu ini berpengaruh pada masukan data pada menu *tallysheet* dimana sejauh ini terdapat 8 (delapan) aktivitas konservasi berdasarkan hasil observasi pada BKSDA Provinsi Bengkulu meliputi : temuan, pelanggaran, kejadian, batas alam, penyuluhan, sosialisasi, koordinasi dan pemasangan papan peringatan. Adapun potongan kode dari menu aktivitas konservasi (*data-bencana.php*) adalah sebagai berikut :

```
<?php
if(isset($_POST['submit'])) {
    $id_bencana=$_POST['id_bencana'];
    $jenis_aktivitas=$_POST['jenis_aktivitas'];

    $nama_bencana=ucwords($_POST['nama_bencana']);
    if(isset($id_bencana))
        $query=mysql_query("UPDATE tbl_bencana SET
        jenis_aktivitas='$jenis_aktivitas',
        nama_bencana='$nama_bencana' WHERE
        id_bencana='$id_bencana'");
    else
        $query=mysql_query("insert into tbl_bencana
        values('$id_bencana','$jenis_aktivitas','$nama_bencana') "
        );
    if($query){
        ?><script language="javascript">alert('Berhasil Input
        Data')</script><?php
        ?><script
        language="javascript">document.location.href="?page=data-
        bencana"</script><?php
    }else{
        echo mysql_error();
    }
}
```

Potongan kode (`$_POST['submit']`) bertujuan untuk melakukan proses pengiriman yang didasarkan pada `id_bencana`, `jenis_aktivitas`, dan `nama_bencana`. Selanjutnya dilakukan fungsi jika (`if`) untuk *update* dan kirim data pada `tbl_bencana` dan diakhiri dengan fungsi `alert pesan`.

### c. Menu Petugas

Menu ini berfungsi sebagai tempat penambahan, penghapusan maupun pengubahan data petugas yang melakukan aktivitas konservasi di lapangan. Sehingga setiap petugas yang melakukan penginputan data dengan aplikasi android haruslah terdaftar dan memiliki akses yang legal. Adapun potongan kode dari menu petugas (*data-petugas2.php*) adalah sebagai berikut :

```
<?php

if(isset($_POST['submit'])){$id_user=$_POST['id_user'];
$username=$_POST['username'];
$password=$_POST['password'];

if(isset($id_user))$query=mysql_query("UPDATE tb_user SET
username='$username', password='$password' WHERE
id_user='$id_user'");

else

$query=mysql_query("insert into tb_user
values('$id_user','$username','$password')");

    if($query){?><script
language="javascript">alert('Berhasil Input
Data')</script><?php

?><script
language="javascript">document.location.href="?page=data-
petugas2"</script><?php

}else{echo mysql_error();

}}else{unset($_POST['submit']);}

}else{unset($_POST['submit']);}
```

Potongan kode (`$_POST['submit']`) menjelaskan proses *input* data petugas berdasarkan *id\_user*, *username* dan *password*. Selanjutnya dilakukan fungsi jika (if) untuk *update* dan kirim data pada *tbl\_user* dan diakhiri dengan fungsi `alert pesan`.

#### d. Menu *Resort*

Menu *Resort* berfungsi sebagai tempat penambahan, penghapusan maupun pengubahan data *resort* atau biasa dikenal dengan wilayah konservasi. Menu *resort* memungkinkan admin untuk menggambar *resort* pada peta yang telah disediakan serta menambahkan informasi terkait secara langsung sehingga *resort* hasil dari penggambaran akan digunakan sebagai acuan pada menu yang menggunakan wilayah konservasi (*resort*) tersebut. Adapun potongan kode dari menu petugas (*data-provinsi.php*) adalah sebagai berikut :

```
drawingManager.setMap(map) ;

google.maps.event.addListener(drawingManager,
'polygoncomplete', function (polygon) {

    var coordinates = polygon.getPath().getArray();

    simpan(coordinates);

});

}else{

    unset($_POST['submit']);

}

}else{

    unset($_POST['submit']);

}
```

Pada potongan kode diatas dilakukan *load drawingManager* terlebih dahulu pada *map*. Selanjutnya `google.maps.event.addListener(drawingManager, 'polygoncomplete', function (polygon)` berfungsi sebagai *manager* untuk membuat *polygon*. Dan proses penyimpanan *polygon* pada fungsi `simpan(coordinates)`.

e. Menu *TallySheet*

Menu *TallySheet* berfungsi sebagai tempat penambahan, penghapusan maupun pengubahan data hasil pengumpulan dari aplikasi android. Pada menu *TallySheet* admin tidak hanya dapat mengelola data masukan dari aplikasi android tetapi juga penambahan data baru yang mungkin terjadi apabila dalam satu kasus aplikasi android yang telah dibuat tidak dapat berjalan dengan baik, menu *TallySheet* dapat dijadikan sarana masukan data yang lain. Pada menu ini, hampir sama dengan struktur *input* data yang ada pada aplikasi android. Adapun potongan kode dari menu *TallySheet* (*info-bencana.php*) sebagai berikut :

```
function ambilForm(){  
    var id_aktivitas=$('#id_aktivitas').val()  
    $.post('ambil_form.php', {id:id_aktivitas },  
function(data){  
        $('#data').html(data)  
    })  
}
```

Pada potongan kode `$.post('ambil_form.php', {id:id_aktivitas }, function(data)` merujuk pada kelas *ambil\_form* yang akan digunakan untuk melakukan pemanggilan kelas *input data* aktivitas konservasi.

Kelas *ambil\_form.php*

```
<?php
$id=$_POST['id'];
switch($id){
case 1: include "info-temuan.php";
break;
case 2: include "info-pelanggaran.php";
break;
case 3: include "info-kejadian.php";
break;
case 4: include "info-batas.php";
break;
case 5: include "info-penyuluhan.php";
break;
case 6: include "info-sosialisasi.php";
break;
case 7: include "info-koordinasi.php";
break;
case 8: include "info-pasangpapan.php";
break;
}
```

Potongan kode `$id=$_POST['id']; switch($id)` menunjukkan pemanggilan atau pergantian *id* berdasarkan *case* yang telah dibuat. Hal itu bertujuan untuk menampilkan halaman yang dipilih berdasarkan *id* nya.



f. Menu Peta

Menu Peta berfungsi sebagai tempat visualisasi hasil dari pengumpulan maupun *input* data baik dari aplikasi mobile maupun dari web. Menu ini tidak hanya menampilkan *marker* atau titik dari lokasi pengambilan data tetapi juga berisikan informasi terkait *marker* tersebut. Disamping itu, menu peta juga terintegrasi dengan peta resort yang telah dibuat pada menu resort sebelumnya. Terdapat fasilitas lain seperti kluster *marker* (*MarkerClusterer*) dimana melakukan pengelompokan *marker* berdasarkan *grid cell* tertentu pada peta. Adapun potongan kode dari menu Peta (*markerer.php*) sebagai berikut :

```
<script type="text/javascript"
src="assets/js/markerclusterer_packed.js"></script>

<script src="assets/js/jquery.js"></script>

<script
src="https://maps.googleapis.com/maps/api/js?v=3.exp&sens
or=false&libraries=geometry"></script>

function peta_awal() {

var lokasibaru = new google.maps.LatLng(-
3.711512879214837,102.51242056488991) ;

    var petaoption = {

        zoom: 8,

        center: lokasibaru,

        mapTypeId: google.maps.MapTypeId.ROADMAP  } ;
```

Potongan kode diatas menunjukan pemanggilan pada javascript *markerclusterer\_packed.js*, *jquery.js* dan peta dari google beserta *drwing toolnya*. Pada awal program dilakukan inisialisasi letak koordinat awal dengan fungsi *google.maps.LatLng*, *zoom* peta, dan tipe penampilan peta.

g. Menu Rute

Menu Rute berfungsi sebagai media untuk menampilkan rute dari masing-masing titik (*marker*) pengambilan data di lapangan berdasarkan koordinat data tersebut. Pada menu ini akan ditampilkan rute dari masing-masing titik ke titik yang lain dan juga tuntutan perjalanan hingga menuju titik tersebut. Tidak hanya rute, jarak tempuh dari masing-masing titik dapat diestimasi. Adapun potongan kode dari menu Rute (*route.php*) sebagai berikut :

```
function calcRoute() {  
  
    var id_kw=$('#id_kawasan').val()  
  
    var id_kw2=$('#id_kawasan2').val()  
  
    $.post("ambil_data_rute.php", {id_kw1:id_kw,  
id_kw2:id_kw2}, function(data){  
  
        var objek=JSON.parse(data)  
  
        start=objek.koordinat[0];  
  
        end=objek.koordinat[1]  
  
        var request = {  
  
            origin:start,  
  
            destination:end,  
  
            travelMode:  
google.maps.DirectionsTravelMode.DRIVING  
  
        };  
  
        directionsService.route(request, function(response,  
status) {  
  
            if (status == google.maps.DirectionsStatus.OK) {  
  
                directionsDisplay.setDirections(response);  
  
            } }); })
```

Potongan kode diatas menunjukan fungsi `calcRoute` (kalkulasi rute) pada peta yang telah dibuat yang merujuk pada kelas `ambil_data_rute.php`. Selanjutnya diinisialisasikan titik awal dan titik akhir berdasarkan data. Fungsi `maps.DirectionsTravelMode.DRIVING` menginisialisasikan tipe penampilan peta yang disertakan mode berkendara (petunjuk arah). Selanjutnya menampilkan direksi menggunakan fungsi `directionsDisplay.setDirections`.

#### *h.* Menu Statistik

Menu Statistik berfungsi sebagai media untuk menampilkan statistik dari pengumpulan data baik yang dilakukan dari aplikasi android maupun dari web berdasarkan *resort* yang ada. Pada pembahasan sebelumnya terdapat penjelasan mengenai banyaknya *resort* yaitu 12 (dua belas) resort yang ada di Provinsi Bengkulu. Pada statistik ini akan digambarkan *resort* mana yang secara empiris melakukan banyak aktivitas konservasi. Tidak hanya itu statistik selanjutnya menggambarkan secara lebih detil para petugas yang melakukan aktivitas konservasi. Sehingga tujuan akhir dari menu Statistika ini adalah mendapat gambaran *resort* aktif dan petugas aktif dalam menggalakan aktivitas konservasi berbasis manajemen resort. Adapun potongan kode dari menu Statistik (*statistik.php*) sebagai berikut :

```

chart = new Highcharts.Chart({
  chart: { renderTo: 'container',      plotBackgroundColor:
null,
          plotBorderWidth: null, plotShadow: false },
  title: { text: 'Grafik Aktivitas Pengirim' },
  tooltip: { formatter: function() {
    return '<b>'+
      this.point.name +'</b>: '+
      Highcharts.numberFormat((this.y/total)*100) +' % ('+
      this.y +' Kejadian)'; }
  },

```

Potongan kode diatas menunjukkan pembuatan objek `chart` menggunakan `Highcharts.Chart` yang bertujuan untuk menampilkan grafik aktivitas pengirim. Selanjutnya fungsi `tooltip` akan menampilkan keterangan yang berisi persentase aktivitas pengirim pada suatu titik dengan fungsi `(this.y/total)*100` .

#### i. Menu Pengaduan

Menu Pengaduan berfungsi sebagai media untuk menampilkan pengaduan dari masyarakat. Menu pengaduan pada sisi admin terfokus pada pengumpulan data aduan masyarakat dari halaman antarmuka user pada menu utama web. Pada menu ini admin dapat melakukan cek aduan yang disampaikan, dalam beberapa kasus masyarakat cenderung lebih aktif dan peka terhadap berbagai macam aktivitas terkait dengan wilayah maupun aktivitas konservasi. Dalam hal ini menu Pengaduan memberikan wadah bagi mereka yang peka terhadap alam dengan memberikan aduan maupun temuan yang terkait. Adapun potongan kode dari menu Pengaduan (*data-pengaduan.php*) sebagai berikut :

```

<?php $query=mysql_query("SELECT * FROM
tb_temuan_masyarakat ORDER BY id_tem_mas");
    while($row=mysql_fetch_assoc($query)){ ?>

<tr><td><?php echo $d=$d+1;?></td>

<td><?php echo $row['id_tem_mas'];?></td>

<td><?php echo $row['nama'];?></td>

<td><?php echo $row['keterangan'];?></td>

<td><?php echo $row['koorx'];?></td>

<td><?php echo $row['koory'];?></td>

<td> <a href="?page=data-
pengaduan&mode=edit&id_tem_mas=<?php echo
$row['id_tem_mas'];?>"> </a>

<a href="?page=data-
pengaduan&mode=delete&id_tem_mas=<?php echo
$row['id_tem_mas'];?>" onClick="return confirm('Apakah
Anda Yakin?')"></a></td>
</tr>

```

Potongan kode diatas menunjukkan penampilan data dengan `$query` yang merujuk pada data pada tabel temuan masyarakat. Selanjutnya disertai dengan pilihan untuk melakukan ubah dan hapus data.

#### j. Menu Logout

Menu *Logout* berfungsi sebagai jalan keluar apabila telah selesai melakukan manajemen baik itu oleh admin maupun oleh operator. Menu *logout* akan secara langsung mengarahkan pengguna pada halaman web utama. Adapun potongan kode dari menu *Logout* (*logout.php*) sebagai berikut :

```

<?php

session_start();

session_destroy();

echo '<META http-equiv="refresh"
content="0;URL=../index.php">' ?>

```

Potongan kode di atas menunjukkan adanya sesi mulai *session\_start()* ; dan sesi selesai *session\_destroy()* ; dilengkapi dengan proses *refresh page* *index.php*.

## 2) Antarmuka *Operator*

Pada antarmuka *operator* terdapat 5 (lima) menu utama yang terdiri dari : beranda, petugas, *tallysheet*, peta dan *logout*. Masing-masing menu memiliki fungsi masing-masing yang terkait baik untuk menampilkan informasi maupun manajemen data. Pada antarmuka *operator* menu yang disajikan sama dengan menu yang terdapat pada antarmuka admin, hanya saja terdapat beberapa menu yang tidak tersedia pada antarmuka *operator* meliputi : menu aktivitas, resort, , rute, statistik dan pengaduan. Penjelasan masing-masing menu pada antarmuka *operator* dapat dilihat pada antarmuka admin (*administrator*) pada penjelasan sebelumnya.

## 3) Antarmuka *Pengguna (User)*

Pada antarmuka *pengguna* terdapat 5 (lima) menu utama yang terdiri dari : beranda, *resort*, aktivitas konservasi, galeri, autentikasi dan 8 (delapan) submenu Seputar Konservasi yang berisi informasi terkait aktivitas konservasi serta submenu lapak pengaduan masyarakat. Masing-masing menu memiliki fungsi menampilkan informasi terkait dengan aktivitas konservasi, dimulai dengan informasi teks, gambar maupun peta. Penjelasan masing-masing menu pada antarmuka *user* dapat dilihat pada penjelasan berikut :

### a. Menu Beranda

Menu Beranda berfungsi sebagai halaman utama yang diakses pada web yang telah dibuat. Menu ini menampilkan informasi terkait latar belakang dan

tujuan penelitian serta beberapa informasi terkait konservasi alam. Adapun potongan kode pada menu Beranda (*index.php*) adalah sebagai berikut :

```
<div id="header" class="container">

<div id="logo"><h1><a href="#">WEBGIS 2013</a></h1></div>

<div id="menu"><ul>

<li class="current_page_item"><a href="#" accesskey="1"
onClick='tampil_peta("beranda") '>BERANDA</a></li>

<li><a href="#resort" accesskey="2"
onClick='tampil_peta("resort") '>RESORT</a></li>

<li><a href="#peta_cluster" accesskey="3"
onClick='tampil_peta("peta_cluster") ' >AKTIVITAS
KONSERVASI</a></li>

<li><a href="#gallery" accesskey="4"
onClick='tampil_peta("foto") '>GALERI</a></li>

<li><a href="./admin" accesskey="5"
title="">AUTENTIFIKASI</a></li></ul></div></div>
```

Potongan kode diatas menjelaskan inisialisasi penampilan *header*, logo dan menu pada *index.php*. Selanjutnya `a href=""`, `accesskey=""` dan metode `onClick` untuk menghasilkan halaman berdasarkan *link* halaman dan kunci akses.

b. Menu *Resort*

Menu *Resort* pada antarmuka pengguna berfungsi untuk menampilkan peta hasil *drawing* oleh admin, sehingga *user* dapat mengetahui wilayah resort yang ada melalui peta yang ditampilkan pada menu *resort* tersebut. Adapun potongan kode pada menu resort adalah sebagai berikut :

```
<li><a href="#resort" accesskey="2"
onClick='tampil_peta("resort") '>RESORT</a></li>
```

Potongan kode diatas `a href="#resort"` menunjukan *link* ke halaman `tampil_peta resort` menggunakan metode `onClick`.

c. Menu Aktivitas Konservasi

Menu Aktivitas Konservasi pada antarmuka pengguna berfungsi untuk menampilkan titik pengambilan data terkait dengan aktivitas konservasi yang dilakukan di lapangan maupun penginputan dari web, tidak hanya menampilkan titik (*marker*) tetapi juga wilayah *resort* serta fasilitas kluster sehingga *user* dapat mengetahui wilayah *resort* yang aktif berdasarkan kluster titik tersebut. Adapun potongan kode pada menu Aktivitas Konservasi adalah sebagai berikut :

```
<li><a href="#peta_cluster" accesskey="3"
onClick='tampil_peta("peta_cluster")' >AKTIVITAS
KONSERVASI</a></li>
```

Potongan kode diatas `a href="#peta_cluster"` menunjukan *link* ke halaman `tampil_peta` `peta_cluster` menggunakan metode `onClick`.

d. Menu Galeri

Menu Galeri pada antarmuka pengguna berfungsi untuk menampilkan foto hasil aktivitas konservasi di lapangan. Foto hasil aktivitas konservasi diencode dengan *base64* dan didecode kembali untuk ditampilkan di web. Adapun potongan kode pada menu Galeri adalah sebagai berikut :

```
$sql="SELECT foto FROM tb_kawasan";
$hasil=mysql_query($sql);$data="<table width='100%'">";
$no=1;while($baris=mysql_fetch_array($hasil)){
    if($baris['foto']!=''){ if($no==4){ $data.="<tr>";
        $no=1; }
        $data.="<td>".<img width='200px'
src='data:image/png;base64,'" . $baris['foto'] . "'></td>";
    if($no==4){ $data.="</tr>";    } $no++; } }echo $data."</table>";
```



Potongan kode `$sql="SELECT foto FROM tb_kawasan";` menunjukkan penampilan foto dari tabel kawasan dengan aturan `if($baris['foto']!=''){if($no==4){ $data."<tr>";` penampilan foto ke-4 pada baris selanjutnya untuk menghasilkan 3 baris galeri foto yang rapi. Selanjutnya pengaturan ukuran foto `img width='200px'` dimana foto yang dihasilkan merupakan *decode* dari **base64**.

#### e. Menu Autentifikasi

Menu Autentifikasi pada antarmuka pengguna berfungsi untuk masuk kedalam antarmuka *administrator* maupun *operator*. Menu autentifikasi akan membimbing *user* pada menu *Login* (*login2.php*) dimana pada menu tersebut diharuskan memasukan *username* dan *password* untuk dapat akses lebih lanjut. Adapun potongan kode pada menu autentifikasi dan menu *login* adalah sebagai berikut :

```
<li><a href="./admin" accesskey="5"
title="">AUTENTIFIKASI</a></li>
```

```
<form name="form" action="cekduku.php" method="POST"
onSubmit="return validasi(this)">

<h1>Login User</h1>      <div id="inputnama">

<input name="username" type="text" placeholder="Username"
/>

</div>

<div id="inputpass"><input name="password"
type="password" placeholder="Password" /> </div>

<div id="aksi"><input type="submit"
value="Login" /> <a href="index.php">Back to WEBGIS?</a>

</div>      </form>
```

Potongan kode `a href="./admin" accesskey="5"` menunjukkan penampilan link pada sesi admin. Selanjutnya dilakukan pengecekan pada kelas `cekduku.php` dengan *username* dan *password* yang dimasukan dengan tipe *submit*.

f. SubMenu Seputar Konservasi

SubMenu Seputar Konservasi pada antarmuka pengguna berfungsi untuk menampilkan informasi terkait aktivitas konservasi. Dalam hal ini terdapat 8 (delapan) submenu seputar konservasi yang disajikan. Adapun potongan kode pada submenu Seputar Konservasi adalah sebagai berikut :

```
<h2>SEPUTAR KONSERVASI</h2></div>

<ul class="style2">      <li><a href="#data_temuan"
onClick='tampil_peta("temuan") '>Temuan Flora dan
Fauna</a></li>

<li><a href="#data_pelanggaran"
onClick='tampil_peta("pelanggaran") '>Pelanggaran
Alam</a></li>

<li><a href="#data_kejadian"
onClick='tampil_peta("kejadian") '>Kejadian Alam</a></li>

<li><a href="#data_batas"
onClick='tampil_peta("batas") '>Tanda Batas
Wilayah</a></li>

<li><a href="#data_penyuluhan" onClick=
'tampil_peta("penyuluhan") '>Penyuluhan
Masyarakat</a></li>

<li><a href="#data_sosialisasi" onClick=
'tampil_peta("sosialisasi") '>Sosialisasi Sosial</a></li>

<li><a href="#data_koordinasi" onClick=
'tampil_peta("koordinasi") '>Koordinasi Sekitar</a></li>

<li><a href="#data_pasangpapan" onClick=
'tampil_peta("pasangpapan") '>Pemasangan Papan
Peringatan</a></li></ul>
```

Potongan kode `a href` menjelaskan penampilan halaman seputar konservasi yang diwakili oleh 8 (delapan) halaman aktivitas konservasi dengan fungsi `onClick`.

g. SubMenu Lapak Pengaduan

SubMenu Lapak Pengaduan pada antarmuka pengguna berfungsi untuk pengumpulan data aduan masyarakat. Dalam beberapa kasus masyarakat cenderung lebih aktif dan peka terhadap berbagai macam aktivitas terkait dengan wilayah maupun aktivitas konservasi. Dalam hal ini submenu Lapak Pengaduan memberikan wadah bagi mereka yang peka terhadap alam dengan memberikan aduan maupun temuan yang terkait. Adapun potongan kode pada submenu Lapak Pengaduan (*temuan\_masyarakat.php*) adalah sebagai berikut :

```
function simpan() {  
  
    var nama = $('input[name=nama]:visible').val();  
  
    var keterangan =  
    $('input[name=keterangan]:visible').val();  
  
    var koorx = $('input[name=koorx]:visible').val();  
  
    var koory = $('input[name=koory]:visible').val();  
  
    $.post("simpan_masyarakat.php", {nama:nama, "keterangan":ke  
    terangan, "koorx":koorx, "koory":koory}, function(data) {
```

Potongan kode diatas menunjukkan fungsi simpan berdasarkan variabel *input* nama, keterangan, koorx, koory pada `$.post` (`"simpan_masyarakat.php"`). Kode pada *temuan\_masyarakat.php* berhubungan dengan *simpan\_masyarakat.php*. Adapun potongan program pada *simpan\_masyarakat.php* adalah sebagai berikut :

```
$nama = $_POST['nama'];  
  
$keterangan = $_POST['keterangan'];  
  
$koorx = $_POST['koorx'];  
  
$koory = $_POST['koory'];  
  
$masuk = mysql_query("insert into  
tb_temuan_masyarakat(nama,keterangan,koorx,koory)  
values('$nama','$keterangan','$koorx','$koory')");
```

## 5.2 Pengujian Sistem

Pengujian sistem dilakukan dalam dua tahap utama, yaitu pengujian *white box* dan *black box*.

### 5.2.1 Pengujian *white box*

Pengujian *white box* dilakukan dengan menguji atribut dan *method* yang ada pada kelas-kelas yang dibangun. Pengujian dilakukan dengan mengecek semua *statement* pada program telah dieksekusi paling tidak satu kali. Pengujian ini dilakukan pada proses pengembangan sistem yakni pengujian kode program (*coding*). Proses pengujian kode program (*coding*) menggunakan *emulator android* yang disebut *Android Virtual Devices (AVD)*.



Gambar 5.10 *Android Virtual Devices (AVD)*

Untuk menjalankan *Android Virtual Devices (AVD)*, *IDE Eclipse* harus telah terinstal *Android SDK*. *Android SDK* adalah *plugin* yang mengkonfigurasi *emulator android*. Berikut pengujian sistem di *Android Virtual Devices (AVD)*.

### 1. Pengujian *Login*

*Login* adalah tampilan awal ketika sistem dijalankan.



Gambar 5.11 *Login (AVD)*

### 2. Pengujian Menu

Menu merupakan tampilan yang muncul setelah *Login*. Di menu ini terdapat menu-menu pilihan yang dapat digunakan oleh *user*. Dimulai dari menu aktivitas konservasi, sinkronisasi, peta dan keluar. Masing-masing menu memiliki fungsi dan keterkaitannya masing-masing.



Gambar 5.12 Menu (AVD)

### 3. Pengujian Menu Aktivitas Konservasi.

Menu Aktivitas Konservasi akan muncul ketika *user* menekan tombol menu Aktivitas Konservasi tersebut yang berada di menu.



Gambar 5.13 Menu Aktivitas Konservasi (AVD)

Pada menu ini terdapat 2 (dua) menu utama yaitu menu Patroli dan menu Penyuluhan dimana masing-masing menu terdapat 4 (empat) submenu. Pada menu Patroli terdapat 4 (empat) submenu yaitu Temuan, Pelanggaran, Kejadian dan Batas. Pada menu Penyuluhan juga terdapat 4 (empat) submenu yaitu Penyuluhan, Sosialisasi, Koordinasi dan Pasangpapan. Berikut merupakan pengujian dari submenu tersebut :



Gambar 5.14 Submenu Temuan (AVD)

Submenu temuan merupakan bagian dari menu Patroli dimana terdapat 2 (dua) objek kajian yaitu Flora dan Fauna, masing-masing kajian tergolong dalam status Dilindungi dan Tidak Dilindungi. Pada submenu ini aplikasi menggunakan `AutoCompleteTextView` untuk melakukan pencarian otomatis status Flora dan Fauna berdasarkan daftar yang disimpan di *database*. Pada *field* koordinat, aplikasi

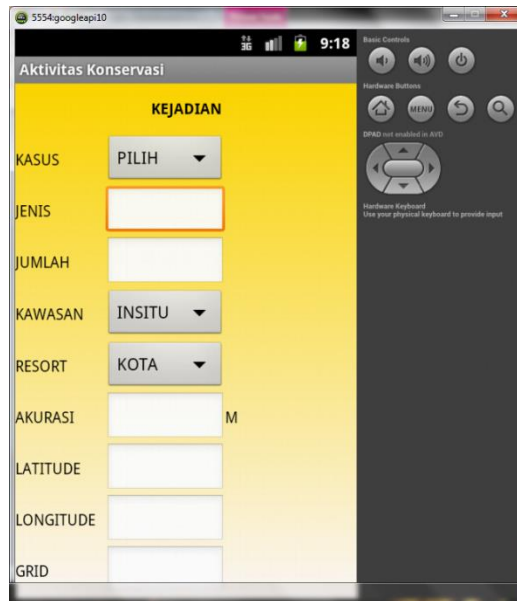
mencari secara langsung koordinat *real time* dari pengguna aplikasi, Koordinat yang dimaksud adalah *Latitude* dan *Longitude* yang akan dimasukan dalam *field* tersebut. Untuk mendukung validitas data, aplikasi juga memungkinkan pengambilan gambar temuan.

The image shows a screenshot of an Android application window titled '5554googleapi10'. The main content area has a yellow background and is titled 'Aktivitas Konservasi' with a subtitle 'PELANGGARAN'. It contains a vertical list of input fields: 'KASUS' with a dropdown menu showing 'ILLEGAL LOGGING', 'JENIS' with an empty text box, 'JUMLAH' with an empty text box, 'KAWASAN' with a dropdown menu showing 'INSITU', 'RESORT' with a dropdown menu showing 'KOTA', 'AKURASI' with an empty text box followed by a unit 'M', 'LATITUDE' with an empty text box, 'LONGITUDE' with an empty text box, and 'GRID' with an empty text box. On the right side, there is a dark grey sidebar with various controls: 'Basic Controls' (volume, power), 'Hardware Buttons' (home, menu, back, search), 'DPAD not enabled in AVD' (a directional pad button), and 'Hardware Keyboard' (a note to use the physical keyboard for input).

Gambar 5.15 Submenu Pelanggaran (AVD)

Pada submenu pelanggaran 5 (lima) objek kajian yaitu Illegal Logging, Perambahan, Perburuan, Pengerusakan dan Kepemilikan. Pada submenu ini hampir sama dengan submenu sebelumnya dimana berisi *field* isian berdasarkan aktivitas di lapangan. Pada *field* ini, aplikasi juga melakukan mencari secara langsung koordinat *Latitude* dan *Longitude* serta memungkinkan pengambilan gambar pelanggaran.





Gambar 5.16 Submenu Kejadian (AVD)

Pada submenu kejadian terdapat 2 (dua) objek kajian yaitu Konflik Satwa dan Titik Api/Kebakaran. Pada submenu ini hampir sama dengan submenu sebelumnya dimana berisi *field* isian berdasarkan aktivitas di lapangan. Pada *field* ini, aplikasi juga melakukan mencari secara langsung koordinat *Latitude* dan *Longitude* serta memungkinkan pengambilan gambar kejadian.



Gambar 5.17 Submenu Tanda Batas (AVD)

Pada submenu kejadian terdapat 3 (tiga) objek kajian yaitu Pal Batas, Batas Alam dan Tanda Lain. Pada submenu ini hampir sama dengan submenu sebelumnya dimana berisi *field* isian berdasarkan aktivitas di lapangan. Pada *field* ini, aplikasi juga melakukan mencari secara langsung koordinat *Latitude* dan *Longitude* serta memungkinkan pengambilan gambar Tanda Batas.



Gambar 5.18 Submenu Penyuluhan (AVD)

Submenu penyuluhan merupakan bagian dari menu Penyuluhan dimana terdapat 2 (dua) objek kajian penyuluhan yaitu Perorangan dan Kelompok. Pada submenu ini aplikasi juga melakukan pencarian otomatis koordinat *Latitude* dan *Longitude* yang akan dimasukan dalam *field* isian. Untuk mendukung validitas data, aplikasi juga memungkinkan pengambilan gambar penyuluhan.



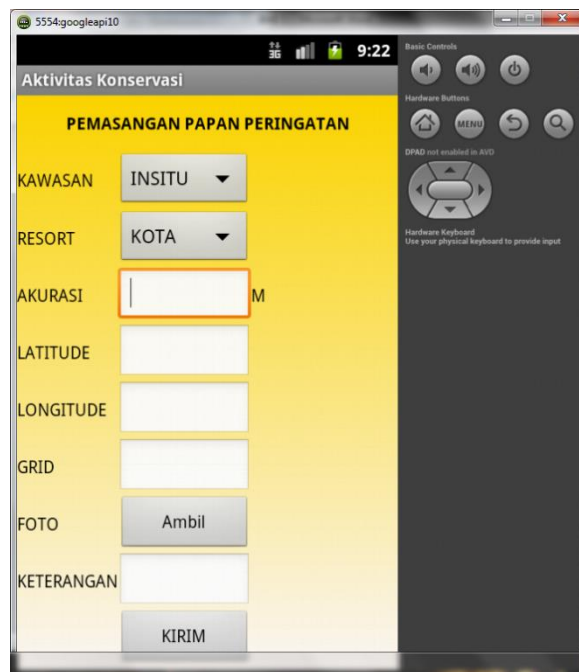
Gambar 5.19 Submenu Sosialisasi (AVD)

Submenu sosialisasi memiliki fungsi yang hampir sama dengan submenu Penyuluhan dimana terdapat 2 (dua) objek kajian sosialisasi yaitu Perorangan dan Kelompok. Pada submenu ini aplikasi juga melakukan pencarian otomatis koordinat *Latitude* dan *Longitude* yang akan dimasukan dalam *field* isian. Untuk mendukung validitas data, aplikasi juga memungkinkan pengambilan gambar sosialisasi.



Gambar 5.20 Submenu Koordinasi (AVD)

Submenu koordinasi memiliki fungsi yang hampir sama dengan submenu sebelumnya dimana terdapat 2 (dua) objek kajian koordinasi yaitu Perorangan dan Kelompok. Pada submenu ini aplikasi juga melakukan pencarian otomatis koordinat *Latitude* dan *Longitude* yang akan dimasukan dalam *field* isian. Untuk mendukung validitas data, aplikasi juga memungkinkan pengambilan gambar koordinasi.

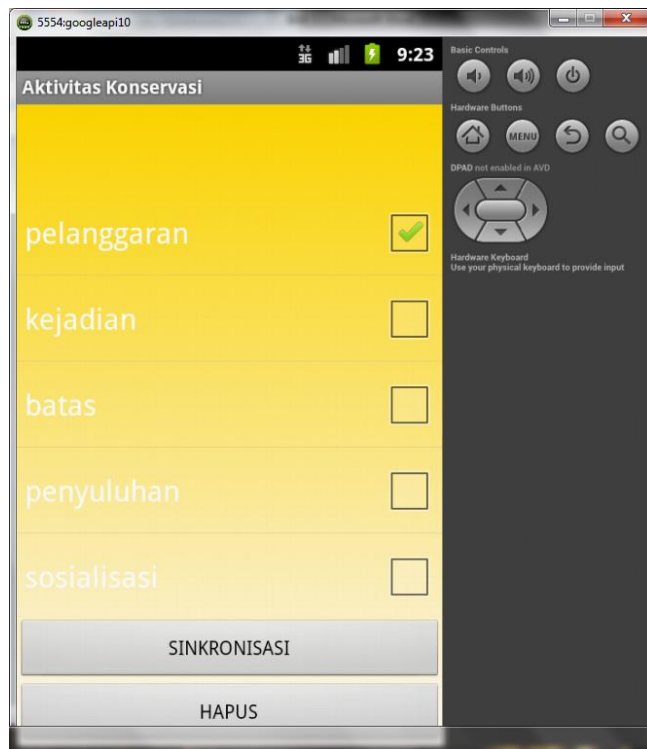


Gambar 5.21 Submenu Pemasangan Papan Peringatan (AVD)

Submenu pemasangan papan peringatan memiliki fungsi yang hampir sama dengan submenu sebelumnya dimana. Terdapat *field* berdasarkan aktivitas di lapangan. Pada submenu ini aplikasi juga melakukan pencarian otomatis koordinat *Latitude* dan *Longitude*. Untuk mendukung validitas data, aplikasi juga memungkinkan pengambilan gambar pemasangan papan peringatan.

#### 4. Pengujian Sinkronisasi

Sinkronisasi merupakan menu tampilan yang muncul setelah melakukan klik menu Sinkronisasi. Pada menu ini mengantisipasi kemungkinan permasalahan teknis yang terjadi di pada proses pengiriman data ke web seperti tidak ada sinyal pengiriman data (*out of range signal*). Menu ini menjamin bahwa setiap aktivitas yang dilakukan oleh masing-masing menu *input* data akan *direcord* kedalam sebuah daftar (*ListView*) sehingga semua data hasil *input* data aktivitas konservasi dapat dimanipulasi untuk dihapus maupun di sinkronisasi kembali.



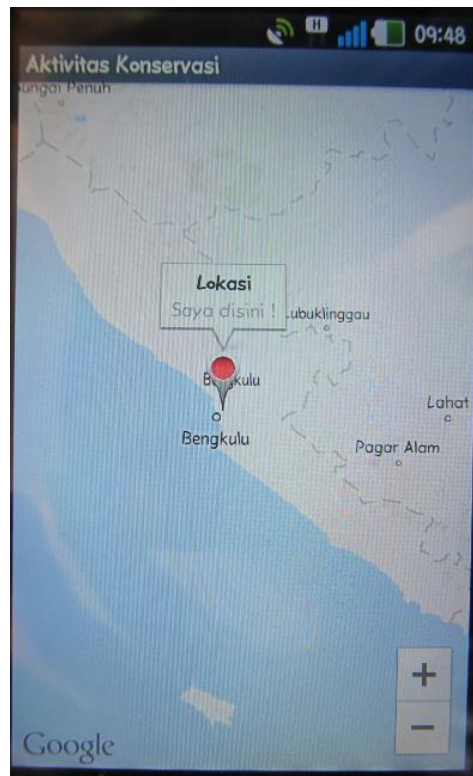
Gambar 5.22 Sinkronisasi (AVD)

Pada menu sinkronisasi ini dilakukan 2 (dua) tahap pengiriman data yang pertama aplikasi akan melakukan pengecekan di web apakah data yang diinputkan sudah ada atau belum. Apabila data telah ada maka akan diberikan notifikasi “Ada” namun apabila data belum ada maka akan diberikan notifikasi “Belum Ada” dan

dilanjutkan dengan sinkronisasi dan notifikasi “Sinkronisasi Berhasil”. Adapun data yang di *CheckList* untuk disinkronisasi dilakukan satu per satu. Terdapat pula menu hapus data berdasarkan *CheckList*.

## 5. Pengujian Peta

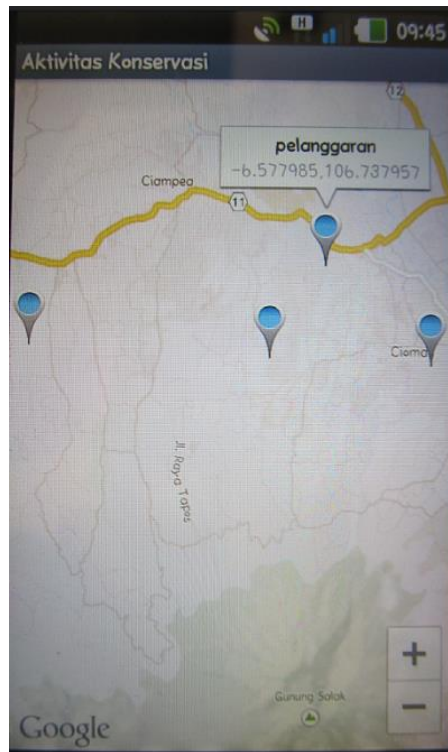
Menu Peta merupakan menu menampilkan peta dimana data koordinat berasal dari *form input* data pada menu aktivitas konservasi sebelumnya. Pada menu ini, peta menampilkan *marker* yang menyatakan titik dan informasi dimana data *input* dimasukan serta lokasi kita saat ini. Penampilan *marker* berfungsi sebagai gambaran petugas dalam pengambilan data di lapangan.



Gambar 5.23 Peta Lokasi *Device* (*Device Android*)

Pada menu Peta menggunakan *Maps API v.2* dimana belum mendukung *running* peta di *emulator* mengingat *v.1* Maps API yang baru saja tidak lagi

digunakan (*deprecated*). Selanjutnya *running program* dijalankan langsung pada *device android*. Pada gambar diatas menunjukan lokasi *device* berada.

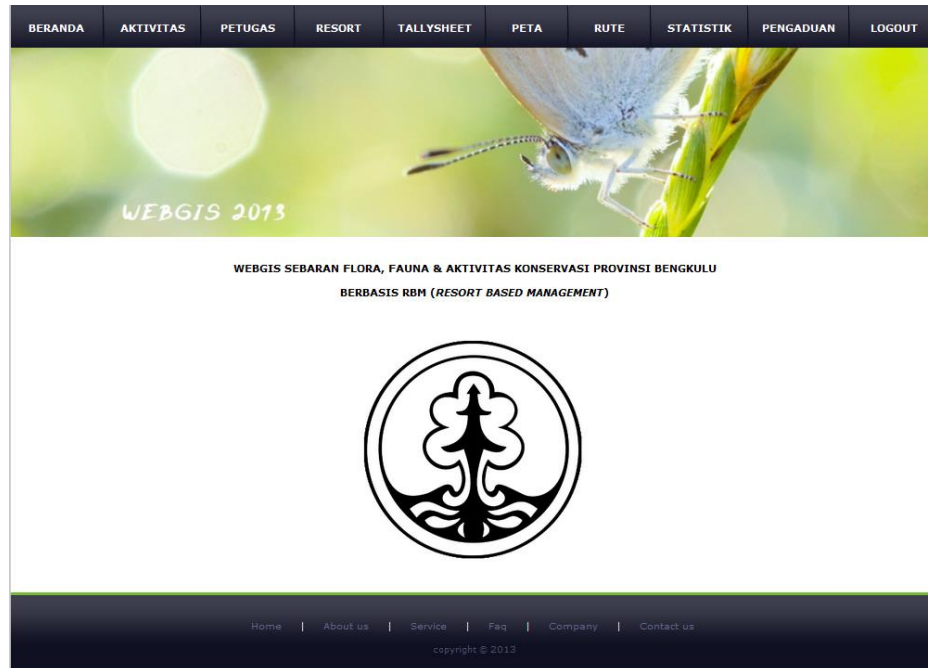


Gambar 5.24 Peta Aktivitas Konservasi (Device Android)

Pada gambar diatas terlihat *marker* yang menunjukan lokasi pengambilan data. Setiap petugas melakukan pengambilan data maka *marker* akan bertambah dengan detail informasi mengenai *marker* tersebut dan koordinat *Latitude* dan *Longitude*.

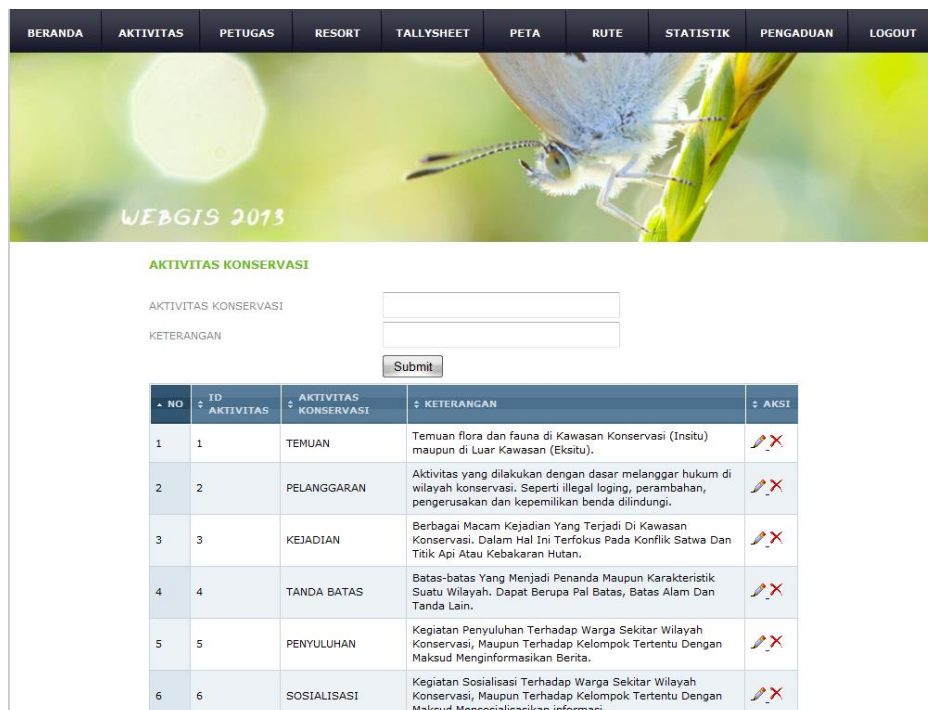
#### 6. Pengujian Antarmuka admin (Administrator) Web

Pada antarmuka admin terdapat 10 (sepuluh) menu utama yang terdiri dari: beranda, aktivitas, petugas, resort, *tallysheet*, peta, rute, statistik, pengaduan dan *logout*. Adapun pengujian dari masing-masing menu akan ditampilkan sebagai berikut :



Gambar 5.25 Beranda (Browser Mozilla Firefox)

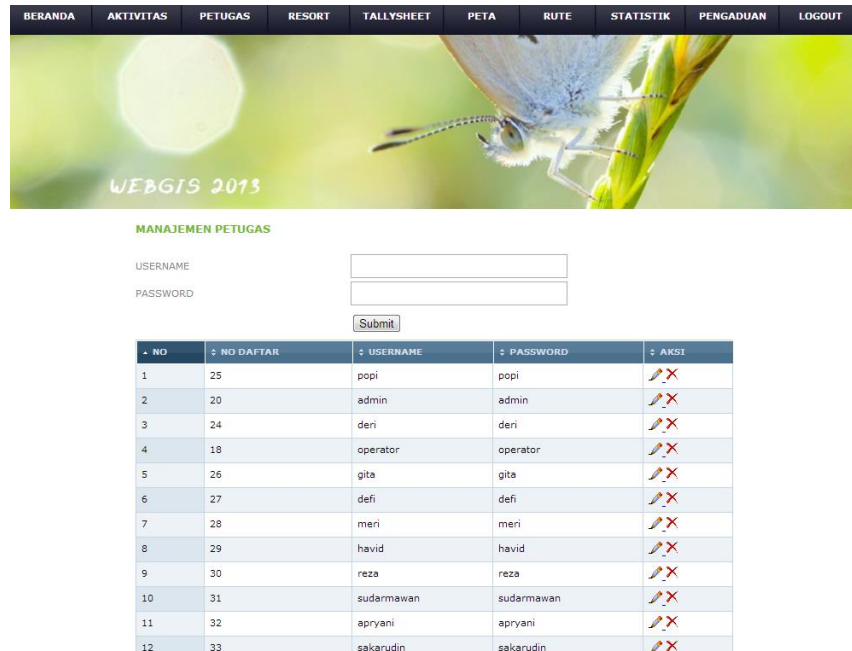
Menu ini berfungsi sebagai antarmuka pertama yang menunjukkan halaman awal admin.



























Gambar 5.26 Aktivitas (Browser Mozilla Firefox)



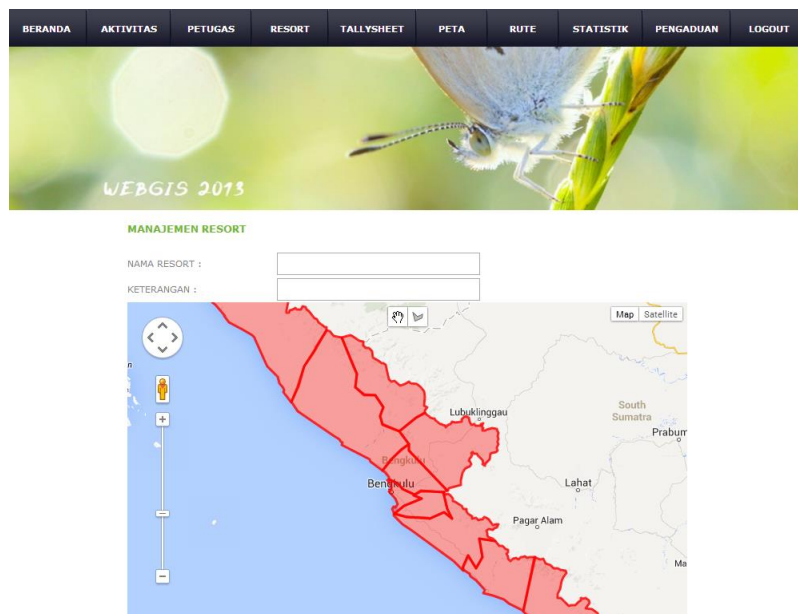
Menu Aktivitas ini berfungsi sebagai tempat penambahan, penghapusan maupun pengubahan aktivitas konservasi yang dilakukan pada penelitian ini, menu ini berpengaruh pada masukan data pada menu *tallysheet*.



NO	NO DAFTAR	USERNAME	PASSWORD	AKSI
1	25	popi	popi	 
2	20	admin	admin	 
3	24	deni	deni	 
4	18	operator	operator	 
5	26	gita	gita	 
6	27	defi	defi	 
7	28	meri	meri	 
8	29	havid	havid	 
9	30	reza	reza	 
10	31	sudarmawan	sudarmawan	 
11	32	apryani	apryani	 
12	33	sakarudin	sakarudin	 

Gambar 5.27 Petugas (Browser Mozilla Firefox)

Menu ini berfungsi sebagai tempat penambahan, penghapusan maupun pengubahan data petugas yang melakukan aktivitas konservasi di lapangan

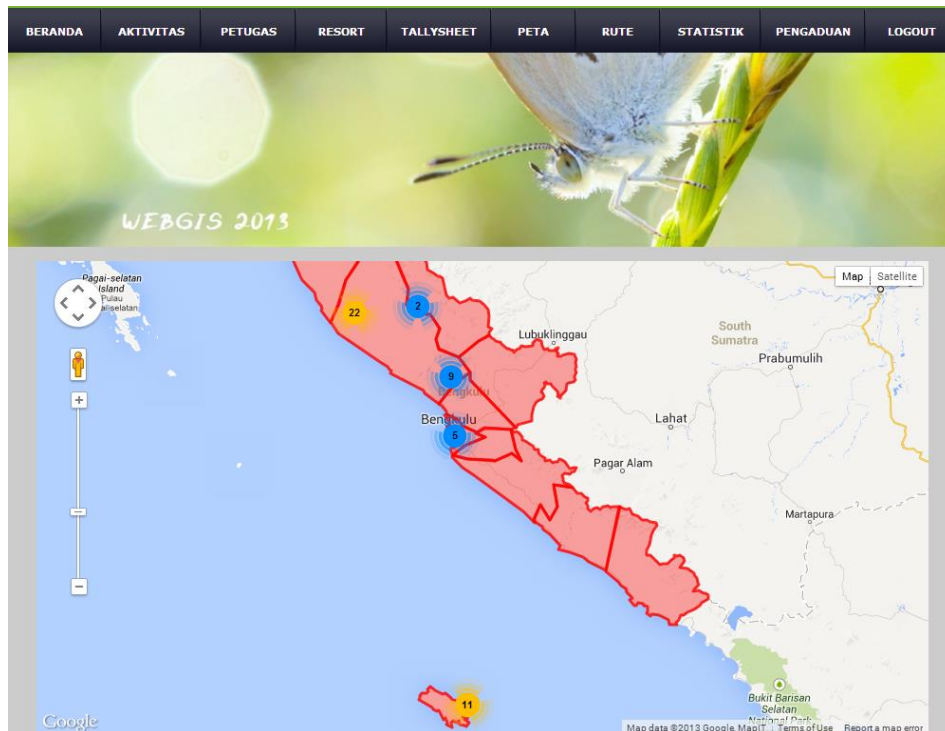


Gambar 5.28 Resort (Browser Mozilla Firefox)

Menu *Resort* berfungsi sebagai tempat penambahan, penghapusan maupun pengubahan data *resort* atau biasa dikenal dengan wilayah konservasi. Menu *resort* memungkinkan admin untuk menggambar *resort* pada peta yang telah disediakan.

Gambar 5.29 *TallySheet* (Browser Mozilla Firefox)

Menu *TallySheet* berfungsi sebagai tempat penambahan, penghapusan maupun pengubahan data hasil pengumpulan dari aplikasi android. Pada menu *TallySheet* admin tidak hanya dapat mengelola data masukan dari aplikasi android tetapi juga penambahan data baru yang mungkin terjadi apabila dalam satu kasus aplikasi android yang telah dibuat tidak dapat berjalan dengan baik, menu *TallySheet* dapat dijadikan sarana masukan data yang lain. Pada menu ini, hampir sama dengan struktur *input* data yang ada pada aplikasi android.



Gambar 5.30 Peta (Browser Mozilla Firefox)

Menu Peta berfungsi sebagai tempat visualisasi hasil dari pengumpulan maupun *input* data baik dari aplikasi mobile maupun dari web. Menu ini tidak hanya menampilkan *marker* atau titik dari lokasi pengambilan data tetapi juga berisikan informasi terkait *marker* tersebut. Disamping itu, menu peta juga terintegrasi dengan peta resort yang telah dibuat pada menu resort sebelumnya. Terdapat fasilitas lain seperti kluster *marker* (*MarkerClusterer*) dimana melakukan pengelompokan *marker* berdasarkan *grid cell* tertentu pada peta.

BERANDA

AKTIVITAS

PETUGAS

RESORT

TALLYSHEET

PETA

RUTE

STATISTIK

PENGADUAN

LOGOUT

WEBGIS 2013

A : TEMUAN-MUKOMUKO

B : TEMUAN-KAUR

Cari Direksi

Map

Satellite

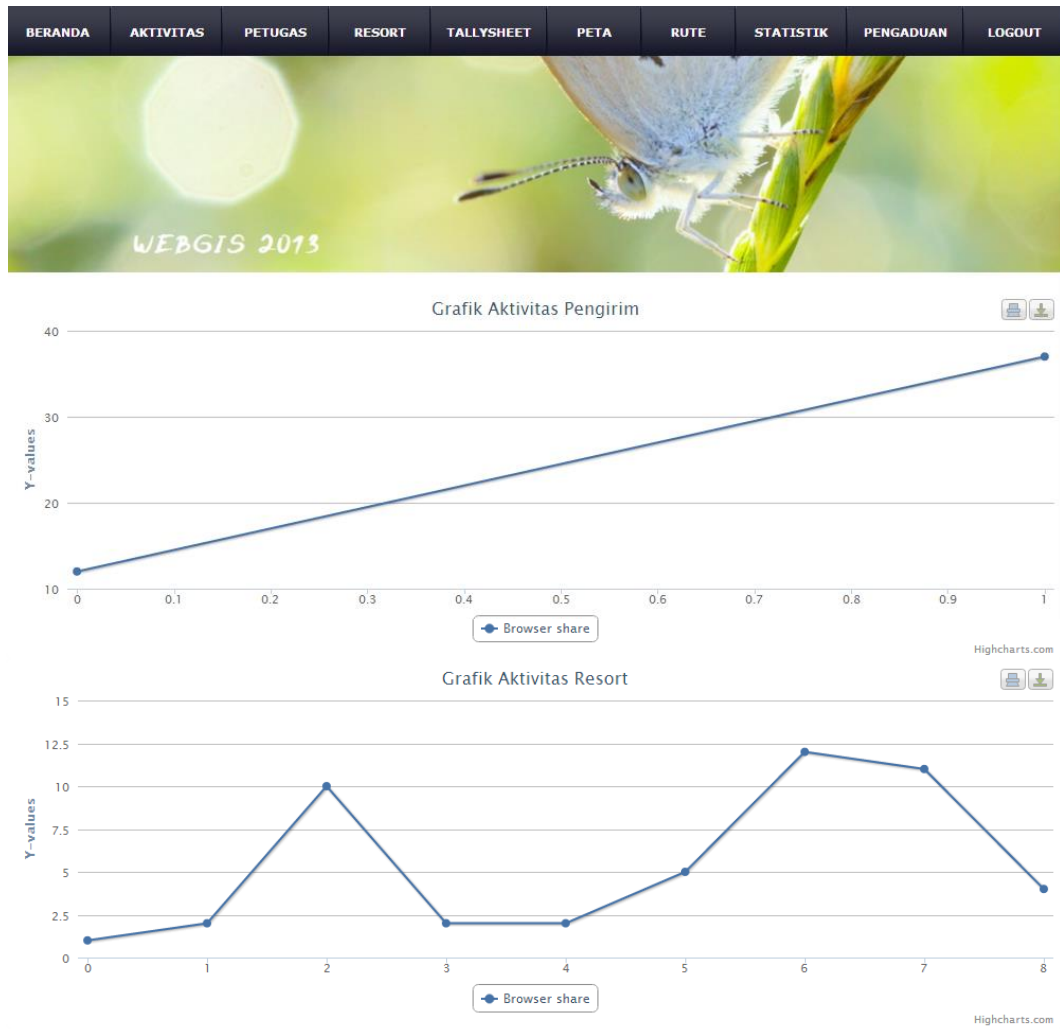
Sidodadi, Rumbai River, Mukomuko, Bengkulu, Republic of Indonesia

263 km - about 5 hours 12 mins

1.	Head <b>north</b>	0.1 km
2.	Turn <b>left</b>	1.7 km
3.	Turn <b>left</b>	0.7 km
4.	Turn <b>right</b>	1.7 km
5.	Turn <b>left</b>	19.0 km
6.	Turn <b>right</b>	18.9 km
7.	Turn <b>right</b>	42.1 km
8.	Turn <b>left</b>	7.1 km
9.	Turn <b>left</b>	2.3 km
10.	At the roundabout, take the <b>3rd</b> exit	57.0 km
11.	Turn <b>right</b>	11.0 km
12.	Turn <b>right</b> toward <b>Jl. Kalimantan</b>	13.2 km
13.	Continue straight onto <b>Jl. Kalimantan</b>	0.6 km
14.	Slight <b>left</b> to stay on <b>Jl. Kalimantan</b>	0.1 km
15.	Continue onto <b>Jl. Sumatra</b>	0.5 km
16.	Turn <b>left</b> to stay on <b>Jl. Sumatra</b>	0.4 km
17.	Continue straight to stay on <b>Jl. Sumatra</b>	0.1 km

Gambar 5.31 *Rute* (Browser Mozilla Firefox)

Menu Rute berfungsi sebagai media untuk menampilkan jarak, estimasi waktu dan rute dari masing-masing titik (*marker*) pengambilan data di lapangan berdasarkan koordinat data tersebut.



Gambar 5.32 *Statistik* (Browser Mozilla Firefox)

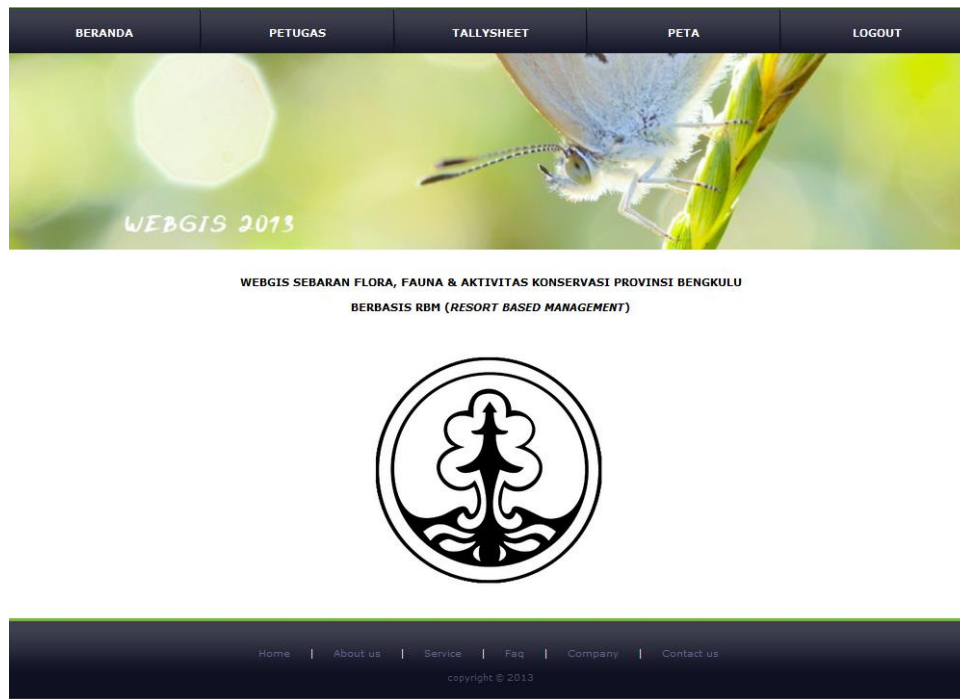
Menu Statistik berfungsi sebagai media untuk menampilkan statistik dari pengumpulan data baik yang dilakukan dari aplikasi android maupun dari web berdasarkan *resort* yang ada. Pada statistik ini akan digambarkan *resort* mana yang secara empiris melakukan banyak aktivitas konservasi. Tidak hanya itu statistik selanjutnya menggambarkan secara lebih detil para petugas yang melakukan aktivitas konservasi. Sehingga memberikan gambaran *resort* aktif dan petugas aktif dalam menggalakan aktivitas konservasi berbasis manajemen resort.

Gambar 5.33 *Pengaduan* (Browser Mozilla Firefox)

Menu Pengaduan berfungsi sebagai media untuk menampilkan pengaduan dari masyarakat. Dalam beberapa kasus masyarakat cenderung lebih aktif dan peka terhadap berbagai macam aktivitas terkait dengan wilayah maupun aktivitas konservasi. Menu Pengaduan memberikan wadah bagi mereka yang peka terhadap alam dengan memberikan aduan maupun temuan yang terkait.

## 7. Pengujian Antarmuka Operator Web

Pada antarmuka *operator* terdapat 5 (lima) menu utama yang terdiri dari : beranda, petugas, *tallysheet*, peta dan *logout*. Pengujian antarmuka *operator* dapat pada penjelasan berikut:



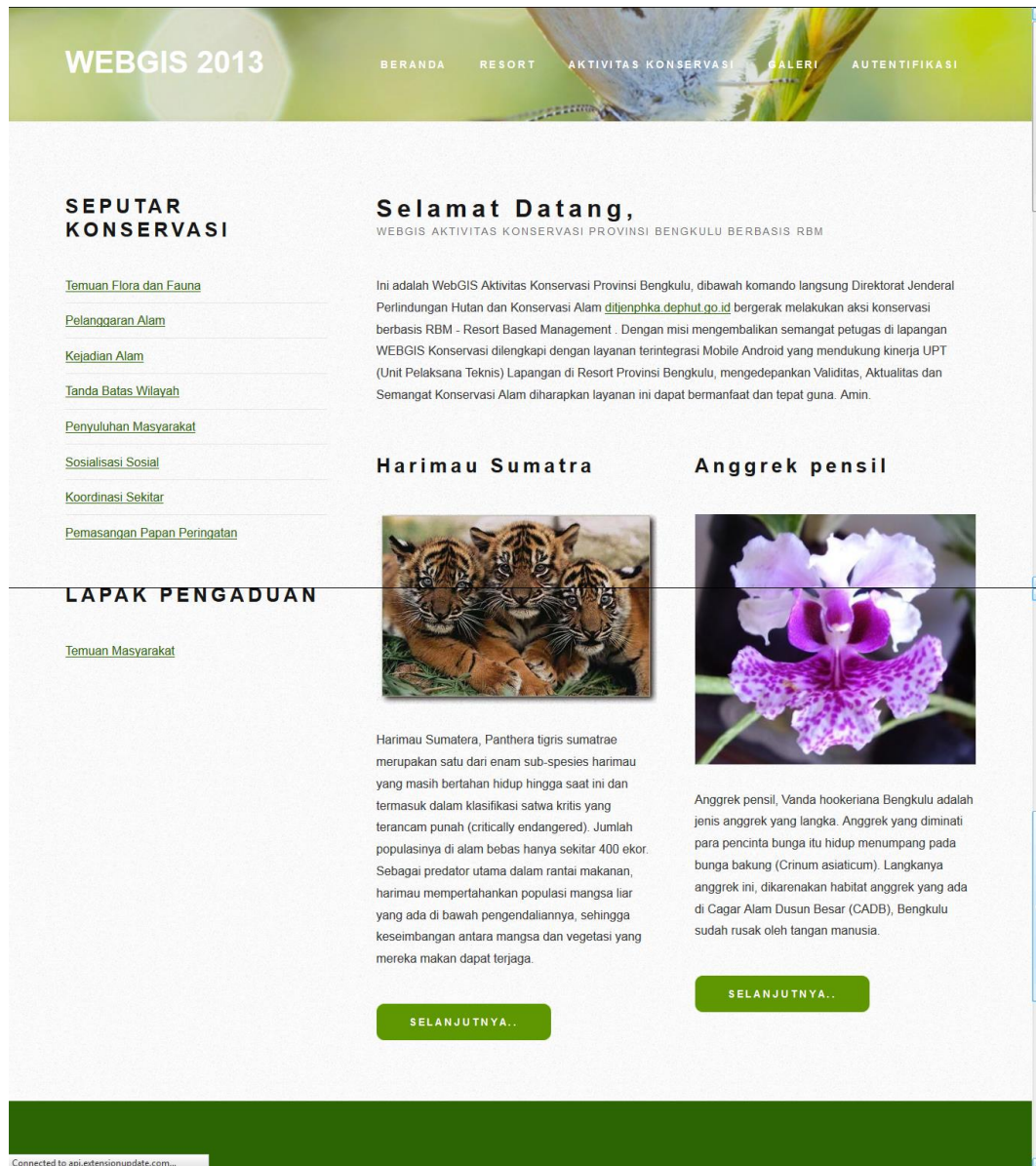
Gambar 5.34 *Antarmuka Operator* (Browser Mozilla Firefox)

Pada antarmuka *operator* menu yang disajikan sama dengan menu yang terdapat pada antarmuka admin, hanya saja terdapat beberapa menu yang tidak tersedia pada antarmuka *operator* meliputi : menu aktivitas, resort, , rute, statistik dan pengaduan.

#### 8. Pengujian Antarmuka Pengguna (*User*)

Antarmuka Pengguna memberikan gambaran informasi umum mengenai aktivitas konservasi yang dilakukan kepada masyarakat. Pengujian antarmuka *User* dapat dilihat pada penjelasan berikut:

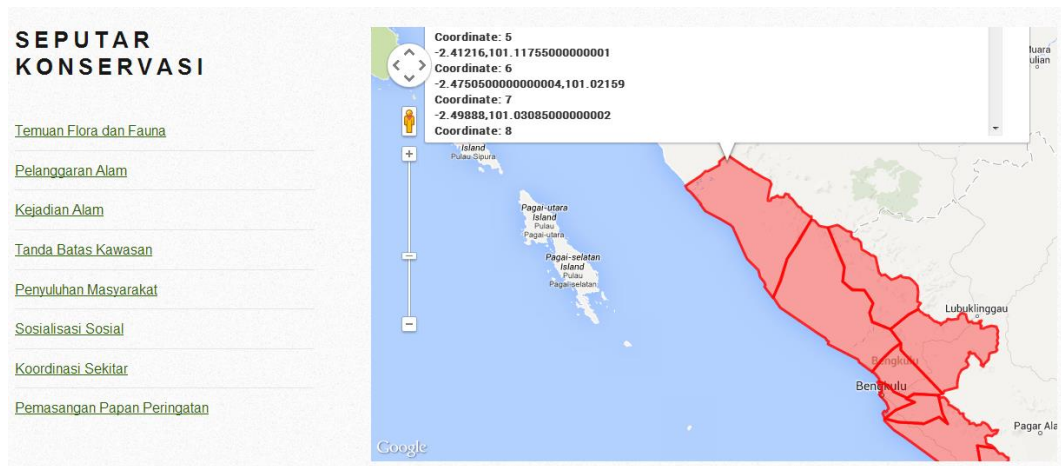




Gambar 5.35 Antarmuka User (Browser Mozilla Firefox)

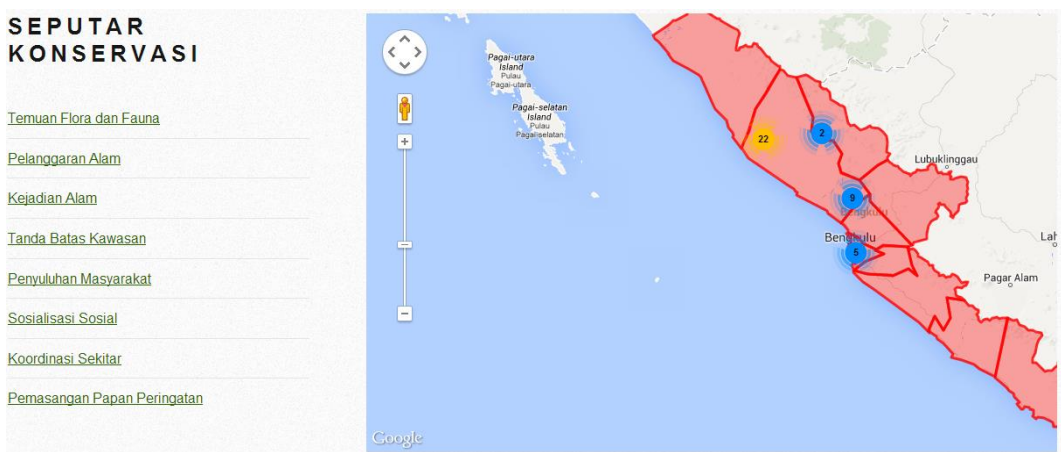
Pada antarmuka *pengguna* terdapat 5 (lima) menu utama yang terdiri dari : beranda, *resort*, aktivitas konservasi, galeri, autentifikasi dan 8 (delapan) submenu Seputar Konservasi yang berisi informasi terkait aktivitas konservasi serta submenu lapak pengaduan masyarakat.





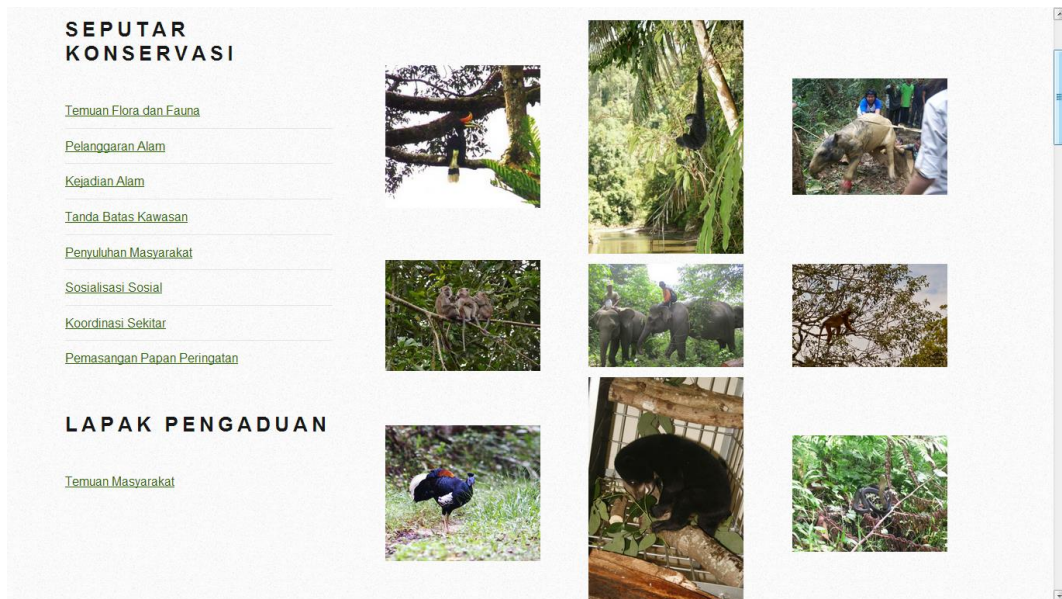
Gambar 5.36 *Menu Resort* (Browser Mozilla Firefox)

Menu *Resort* pada antarmuka pengguna berfungsi untuk menampilkan peta hasil *drawing resort* oleh admin, sehingga *user* dapat mengetahui wilayah resort yang ada melalui peta yang ditampilkan pada menu *resort* tersebut.



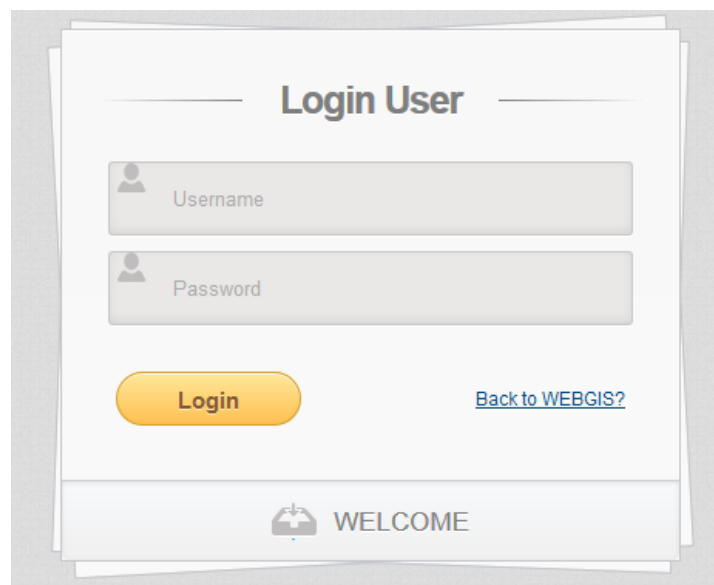
Gambar 5.37 *Menu Aktivitas Konservasi* (Browser Mozilla Firefox)

Menu *Aktivitas Konservasi* pada antarmuka pengguna berfungsi untuk menampilkan titik pengambilan data terkait dengan aktivitas konservasi yang dilakukan di lapangan maupun penginputan dari web, tidak hanya menampilkan titik (*marker*) tetapi juga wilayah *resort* serta fasilitas kluster sehingga *user* dapat mengetahui wilayah *resort* yang aktif berdasarkan kluster titik tersebut.



Gambar 5.38 *Menu Galeri* (Browser Mozilla Firefox)

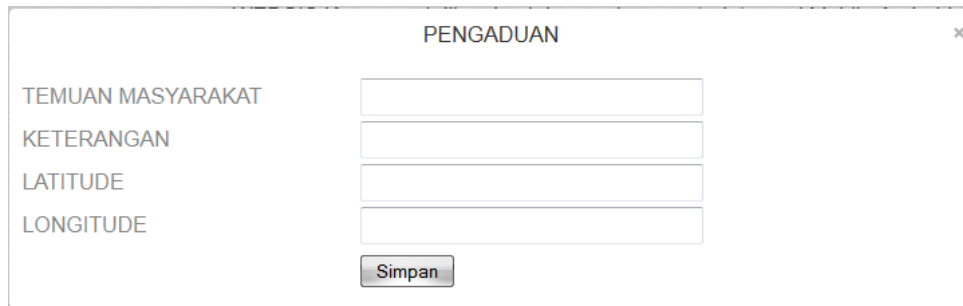
Menu Galeri pada antarmuka pengguna berfungsi untuk menampilkan foto hasil aktivitas konservasi di lapangan. Foto hasil aktivitas konservasi diencode dengan *base64* dan didecode kembali untuk ditampilkan di web.



Gambar 5.39 *Menu Autentifikasi* (Browser Mozilla Firefox)

Menu Autentifikasi pada antarmuka pengguna berfungsi untuk masuk kedalam antarmuka *administrator* maupun *operator*. Menu autentifikasi akan

membimbing *user* pada menu *Login* (*login2.php*) dimana pada menu tersebut diharuskan memasukan *username* dan *password* untuk dapat akses lebih lanjut.



Gambar 5.40 SubMenu Lapak Pengaduan (Browser Mozilla Firefox)

SubMenu Lapak Pengaduan pada antarmuka pengguna berfungsi untuk pengumpulan data aduan masyarakat. Dalam beberapa kasus masyarakat cenderung lebih aktif dan peka terhadap berbagai macam aktivitas terkait dengan wilayah maupun aktivitas konservasi.



**SEPUTAR KONSERVASI**

- [Temuan Flora dan Fauna](#)
- [Pelanggaran Alam](#)
- [Kejadian Alam](#)
- [Tanda Batas Wilayah](#)
- [Penyuluhan Masyarakat](#)
- [Sosialisasi Sosial](#)
- [Koordinasi Sekitar](#)
- [Pemasangan Papan Peringatan](#)

**LAPAK PENGADUAN**

- [Temuan Masyarakat](#)

### Harimau Sumatra



Harimau Sumatera, *Panthera tigris sumatrae* merupakan satu dari enam sub-spesies harimau yang masih bertahan hidup hingga saat ini dan termasuk dalam klasifikasi satwa kritis yang terancam punah (*critically endangered*). Jumlah populasinya di alam bebas hanya sekitar 400 ekor. Sebagai predator utama dalam rantai makanan, harimau mempertahankan populasi mangsa liar yang ada di bawah pengendaliannya, sehingga keseimbangan antara mangsa dan vegetasi yang mereka makan dapat terjaga.

Gambar 5.41 SubMenu Seputar Konservasi (Browser Mozilla Firefox)

SubMenu Seputar Konservasi pada antarmuka pengguna berfungsi untuk menampilkan informasi terkait aktivitas konservasi. Dalam hal ini terdapat 8 (delapan) submenu seputar konservasi yang disajikan.

### 5.2.2 Pengujian *black box*

Pengujian *black box* dilakukan untuk menguji apakah sistem yang dikembangkan sesuai dengan apa yang tertuang dalam spesifikasi fungsional sistem. *Black box* juga digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang dirancang. Kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran yang dihasilkan dari data atau kondisi masukan yang diberikan untuk fungsi yang ada tanpa melihat bagaimana proses untuk mendapatkan keluaran tersebut.

Dari keluaran yang dihasilkan, kemampuan program dalam memenuhi kebutuhan pemakai dapat diukur sekaligus dapat diketahui kesalahan-kesalahannya. Beberapa jenis kesalahan yang dapat diidentifikasi: fungsi tidak benar atau hilang, kesalahan antar muka, kesalahan pada struktur data *database*, dan kesalahan performasi.

#### 1. Pengujian Fungsional Sistem (Aplikasi Android)

##### 1) Masuk Aplikasi

Kasus dan hasil uji (Normal)

Tabel 5.1 Pengujian Masuk Aplikasi

Aktivitas Pengujian	Realisasi yang diharapkan	Hasil
Klik icon dengan judul “Aktivitas Konservasi” pada <i>smartphone</i> Android.	Masuk kedapam aplikasi serta menampilkan menu <i>Login</i> untuk autentifikasi pengguna.	Sukses

## 2) Menu Login

Kasus dan hasil uji (Normal)

Tabel 5.2 Pengujian Beranda Menu

Aktivitas Pengujian	Realisasi yang diharapkan	Hasil
Menampilkan <i>field</i> isian <i>username</i> dan <i>password</i>	Muncul <i>field</i> isian <i>username</i> dan <i>password</i>	Sukses
Klik tombol <i>Login</i>	Muncul tampilan menu utama	Sukses
Klik tombol Keluar	Keluar dari aplikasi	Sukses
Klik tombol Skip	Muncul tampilan menu utama tanpa harus melakukan autentifikasi.	Sukses
Memasukan <i>username</i> dan <i>password</i> secara salah	Muncul kotak dialog gagal <i>login</i>	sukses

## 3) Menu Utama

Kasus dan hasil uji (Normal)

Tabel 5.3 Pengujian Menu Utama (uji normal)

Aktivitas Pengujian	Realisasi yang diharapkan	Hasil
Menampilkan 4 (empat) pilihan utama yaitu Aktivitas Konservasi, Sinkronisasi, Peta dan Keluar	Muncul 4 (empat) pilihan utama yaitu Aktivitas Konservasi, Sinkronisasi, Peta dan Keluar	Sukses
Klik <i>button</i> Aktivitas Konservasi	Menampilkan menu Aktivitas Konservasi	Sukses
Klik <i>button</i> Sinkronisasi	Menampilkan menu sinkronisasi	Sukses
Klik <i>button</i> Peta	Menampilkan Peta titik hasil pengumpulan data	Sukses
Klik <i>button</i> Keluar	Kembali ke menu <i>Login</i>	Sukses

#### 4) Menu Aktivitas Konservasi

Kasus dan hasil uji (Normal)

Tabel 5.4 Pengujian Menu Aktivitas Konservasi

<b>Aktivitas Pengujian</b>	<b>Realisasi yang diharapkan</b>	<b>Hasil</b>
Menampilkan 2 (dua) pilihan utama yaitu Patroli dan Penyuluhan	Muncul 2 (dua) pilihan utama yaitu Patroli dan Penyuluhan	Sukses
Klik menu Patroli	Muncul halaman menu Patroli	Sukses
Klik menu Penyuluhan	Muncul halaman menu Penyuluhan	Sukses

#### 5) Menu Patroli

Kasus dan hasil uji (Normal)

Tabel 5.5 Pengujian Menu Patroli

<b>Aktivitas Pengujian</b>	<b>Realisasi yang diharapkan</b>	<b>Hasil</b>
Menampilkan 4 (empat) pilihan utama yaitu Temuan, Pelanggaran, Kejadian dan Tanda Batas	Muncul 4 (empat) pilihan utama yaitu Temuan, Pelanggaran, Kejadian dan Tanda Batas	Sukses
Klik menu Temuan	Menampilkan halaman isian Temuan	Sukses
Klik menu Pelanggaran	Menampilkan halaman isian Pelanggaran	Sukses
Klik menu Kejadian	Menampilkan halaman isian Kejadian	Sukses
Klik menu Tanda Batas	Menampilkan halaman isian Tanda Batas	Sukses

6) Menu Penyuluhan

Kasus dan hasil uji (Normal)

Tabel 5.6 Pengujian Menu Penyuluhan

Aktivitas Pengujian	Realisasi yang diharapkan	Hasil
Menampilkan 4 (empat) pilihan utama yaitu Penyuluhan, Sosialisasi, Koordinasi dan Pemasangan Papan Peringatan	Muncul 4 (empat) pilihan utama yaitu Penyuluhan, Sosialisasi, Koordinasi dan Pemasangan Papan Peringatan	Sukses
Klik menu Penyuluhan	Menampilkan halaman isian Penyuluhan	Sukses
Klik menu Sosialisasi	Menampilkan halaman isian Sosialisasi	Sukses
Klik menu Koordinasi	Menampilkan halaman isian Koordinasi	Sukses
Klik menu Pemasangan Papan Peringatan	Menampilkan halaman isian Pemasangan Papan Peringatan	Sukses

7) Submenu Temuan

Kasus dan hasil uji (Normal)

Tabel 5.7 Pengujian Submenu Temuan (uji normal)

Aktivitas Pengujian	Realisasi yang diharapkan	Hasil
Menampilkan halaman isian Temuan	Muncul menu isian Temuan	Sukses
Uji <i>Scroll</i> halaman isian Temuan	Halaman dapat <i>Scroll Touch</i> ke bawah maupun keatas.	Sukses
Uji <i>auto complete text</i> pada <i>field</i>	<i>Field</i> dapat menampilkan <i>auto complete text</i>	Sukses



Uji <i>Enable/Disable field</i>	<i>Field</i> dapat diaktifkan dan di non aktifkan berdasarkan pilihan	Sukses
Uji <i>auto generate</i> koordinat <i>Latitude</i> dan <i>Longitude</i>	<i>Field</i> dapat menampilkan koordinat <i>Latitude</i> dan <i>Longitude</i> secara otomatis	Sukses
Uji <i>button</i> ambil gambar	Menampilkan menu ambil gambar	Sukses
Klik <i>button</i> kirim	Melakukan pengiriman data isian ke web	Sukses

#### 8) Submenu Pelanggaran

Kasus dan hasil uji (Normal)

Tabel 5.8 Pengujian Submenu Pelanggaran

<b>Aktivitas Pengujian</b>	<b>Realisasi yang diharapkan</b>	<b>Hasil</b>
Menampilkan halaman isian Pelanggaran	Muncul menu isian Pelanggaran	Sukses
Uji <i>Scroll</i> halaman isian Temuan	Halaman dapat <i>Scroll Touch</i> ke bawah maupun keatas.	Sukses
Uji <i>Enable/Disable field</i>	<i>Field</i> dapat diaktifkan dan di non aktifkan berdasarkan pilihan	Sukses
Uji <i>auto generate</i> koordinat <i>Latitude</i> dan <i>Longitude</i>	<i>Field</i> dapat menampilkan koordinat <i>Latitude</i> dan <i>Longitude</i> secara otomatis	Sukses
Uji <i>button</i> ambil gambar	Menampilkan menu ambil gambar	Sukses
Klik <i>button</i> kirim	Melakukan pengiriman data isian ke web	Sukses



9) Submenu Kejadian

Kasus dan hasil uji (Normal)

Tabel 5.9 Pengujian Submenu Kejadian

Aktivitas Pengujian	Realisasi yang diharapkan	Hasil
Menampilkan halaman isian Kejadian	Muncul menu isian Kejadian	Sukses
Uji <i>Scroll</i> halaman isian Kejadian	Halaman dapat <i>Scroll Touch</i> ke bawah maupun keatas.	Sukses
Uji <i>Enable/Disable field</i>	<i>Field</i> dapat diaktifkan dan di non aktifkan berdasarkan pilihan	Sukses
Uji <i>auto generate</i> koordinat <i>Latitude</i> dan <i>Longitude</i>	<i>Field</i> dapat menampilkan koordinat <i>Latitude</i> dan <i>Longitude</i> secara otomatis	Sukses
Uji <i>button</i> ambil gambar	Menampilkan menu ambil gambar	Sukses
Klik <i>button</i> kirim	Melakukan pengiriman data isian ke web	Sukses

10) Menu Submenu Tanda Batas

Kasus dan hasil uji (Normal)

Tabel 5.10 Pengujian Submenu Tanda Batas

Aktivitas Pengujian	Realisasi yang diharapkan	Hasil
Menampilkan halaman isian Tanda Batas	Muncul menu isian Tanda Batas	Sukses
Uji <i>Scroll</i> halaman isian Tanda Batas	Halaman dapat <i>Scroll Touch</i> ke bawah maupun keatas.	Sukses
Uji <i>Enable/Disable field</i>	<i>Field</i> dapat diaktifkan dan di non aktifkan berdasarkan pilihan	Sukses

Uji <i>auto generate</i> koordinat <i>Latitude</i> dan <i>Longitude</i>	<i>Field</i> dapat menampilkan koordinat <i>Latitude</i> dan <i>Longitude</i> secara otomatis	Sukses
Uji <i>button</i> ambil gambar	Menampilkan menu ambil gambar	Sukses
Klik <i>button</i> kirim	Melakukan pengiriman data isian ke web	Sukses

#### 11) Menu Submenu Penyuluhan

Kasus dan hasil uji (Normal)

Tabel 5.11 Pengujian Submenu Penyuluhan

Aktivitas Pengujian	Realisasi yang diharapkan	Hasil
Menampilkan halaman isian Penyuluhan	Muncul menu isian Penyuluhan	Sukses
Uji <i>Scroll</i> halaman isian Penyuluhan	Halaman dapat <i>Scroll Touch</i> ke bawah maupun keatas.	Sukses
Uji <i>Enable/Disable field</i>	<i>Field</i> dapat diaktifkan dan di non aktifkan berdasarkan pilihan	Sukses
Uji <i>auto generate</i> koordinat <i>Latitude</i> dan <i>Longitude</i>	<i>Field</i> dapat menampilkan koordinat <i>Latitude</i> dan <i>Longitude</i> secara otomatis	Sukses
Uji <i>button</i> ambil gambar	Menampilkan menu ambil gambar	Sukses
Klik <i>button</i> kirim	Melakukan pengiriman data isian ke web	Sukses

12) Menu Submenu Sosialisasi

Kasus dan hasil uji (Normal)

Tabel 5.12 Pengujian Submenu Sosialisasi

Aktivitas Pengujian	Realisasi yang diharapkan	Hasil
Menampilkan halaman isian Sosialisasi	Muncul menu isian Sosialisasi	Sukses
Uji <i>Scroll</i> halaman isian Sosialisasi	Halaman dapat <i>Scroll Touch</i> ke bawah maupun keatas.	Sukses
Uji <i>Enable/Disable field</i>	<i>Field</i> dapat diaktifkan dan di non aktifkan berdasarkan pilihan	Sukses
Uji <i>auto generate</i> koordinat <i>Latitude</i> dan <i>Longitude</i>	<i>Field</i> dpaat menampilkan koordinat <i>Latitude</i> dan <i>Longitude</i> secara otomatis	Sukses
Uji <i>button</i> ambil gambar	Menampilkan menu ambil gambar	Sukses
Klik <i>button</i> kirim	Melakukan pengiriman data isian ke web	Sukses

13) Menu Submenu Koordinasi

Kasus dan hasil uji (Normal)

Tabel 5.13 Pengujian Submenu Koordinasi

Aktivitas Pengujian	Realisasi yang diharapkan	Hasil
Menampilkan halaman isian Koordinasi	Muncul menu isian Koordinasi	Sukses
Uji <i>Scroll</i> halaman isian Koordinasi	Halaman dapat <i>Scroll Touch</i> ke bawah maupun keatas.	Sukses
Uji <i>Enable/Disable field</i>	<i>Field</i> dapat diaktifkan dan di non aktifkan berdasarkan pilihan	Sukses

Uji <i>auto generate</i> koordinat <i>Latitude</i> dan <i>Longitude</i>	<i>Field</i> dapat menampilkan koordinat <i>Latitude</i> dan <i>Longitude</i> secara otomatis	Sukses
Uji <i>button</i> ambil gambar	Menampilkan menu ambil gambar	Sukses
Klik <i>button</i> kirim	Melakukan pengiriman data isian ke web	Sukses

#### 14) Menu Submenu Pemasangan Papan Peringatan

Kasus dan hasil uji (Normal)

Tabel 5.14 Pengujian Submenu Pemasangan Papan Peringatan

<b>Aktivitas Pengujian</b>	<b>Realisasi yang diharapkan</b>	<b>Hasil</b>
Menampilkan halaman isian Pemasangan Papan Peringatan	Muncul menu isian Pemasangan Papan Peringatan	Sukses
Uji <i>Scroll</i> halaman isian Pemasangan Papan Peringatan	Halaman dapat <i>Scroll Touch</i> ke bawah maupun keatas.	Sukses
Uji <i>Enable/Disable field</i>	<i>Field</i> dapat diaktifkan dan di non aktifkan berdasarkan pilihan	Sukses
Uji <i>auto generate</i> koordinat <i>Latitude</i> dan <i>Longitude</i>	<i>Field</i> dapat menampilkan koordinat <i>Latitude</i> dan <i>Longitude</i> secara otomatis	Sukses
Uji <i>button</i> ambil gambar	Menampilkan menu ambil gambar	Sukses
Klik <i>button</i> kirim	Melakukan pengiriman data isian ke web	Sukses

15) Menu Sinkronisasi

Kasus dan hasil uji (Normal)

Tabel 5.15 Pengujian Sinkronisasi

Aktivitas Pengujian	Realisasi yang diharapkan	Hasil
Menampilkan halaman Sinkronisasi	Muncul halaman Sinkronisasi	Sukses
Uji <i>ListView</i> data	Halaman dapat menampilkan <i>ListView</i> data.	Sukses
Uji <i>Check Box</i> pada <i>ListView</i>	<i>Check Box</i> dapat ditampilkan pada <i>ListView</i>	Sukses
Uji <i>button Sinkronisasi</i>	Data pada <i>Listview</i> dapat di sinkronisasi ke web	Sukses
Uji <i>button</i> hapus	Menghapus data pada <i>ListView</i>	Sukses

16) Menu Peta

Kasus dan hasil uji (Normal)

Tabel 5.16 Pengujian Peta

Aktivitas Pengujian	Realisasi yang diharapkan	Hasil
Menampilkan halaman Peta	Muncul halaman Peta	Sukses
Uji <i>OnLoad</i> Peta	Halaman dapat menampilkan Peta	Sukses
Uji menampilkan <i>marker</i> data hasil aktivitas konservasi	<i>Marker</i> data hasil aktivitas konservasi dapat ditampilkan	Sukses
Uji menampilkan <i>current position marker</i>	<i>Current position marker</i> dapat ditampilkan.	Sukses
Uji menampilkan informasi <i>marker</i> koordinat <i>Latitude</i> dan <i>Longitude</i>	Informasi <i>marker</i> koordinat <i>Latitude</i> dan <i>Longitude</i> dapat ditampilkan.	Sukses

17) Pengujian Fungsional Sistem (Web)

Tabel 5.17 Pengujian Web

<b>Aktivitas Pengujian</b>	<b>Realisasi yang diharapkan</b>	<b>Hasil</b>
Tampil halaman pengguna ( <i>user</i> )	Mampu menampilkan informasi umum terkait Aktivitas Konservasi, Peta Hasil dan Galeri	Sukses
Autentifikasi Pengguna ( <i>Administrator</i> dan <i>Operator</i> )	Masuk ke halaman autentifikasi. Melakukan akses <i>Administrator</i> dan <i>Operator</i>	Sukses
Tampil halaman <i>Administrator</i>	Masuk ke halaman <i>Administrator</i> . Melakukan manajemen data aktivitas konservasi, menghasilkan peta dan statistik hasil aktivitas.	Sukses
Tampil halaman <i>Operator</i>	Masuk ke halaman <i>Operator</i> Melakukan manajemen data berdasarkan hak akses <i>operator</i>	Sukses

### 5.3 Uji Kelayakan Sistem

Pengujian kelayakan sistem bertujuan mendapatkan penilaian langsung terhadap sistem yang dihasilkan. Target dari pengujian kelayakan sistem ini adalah responden (calon pemakai sistem). tahapan dari uji kelayakan ini adalah :

#### 1. Angket

Angket yang dibuat berisikan pertanyaan-pertanyaan berkaitan dengan sistem yang dihasilkan. Pertanyaan yang berdasarkan beberapa variabel yaitu : tampilan, kemudahan pengguna dan kinerja dari sistem. Variabel yang ada dijadikan sebagai titik tolak untuk menyusun item-item instrument yang dapat berupa pertanyaan. Penyusunan bentuk jawaban dari pertanyaan menggunakan skala *likert*. Skala *Likert* disebut *Summented Rating Scale*. Ciri khas dari skala ini adalah bentuk jawaban dari pertanyaan mempunyai gradasi sangat positif sampai sangat negatif. Untuk Angket penelitian ini diberikan gradasi jawaban : SB = (Sangat Baik) ; B = (Baik) ; CB = (Cukup Baik) ; KB = (Kurang Baik) ; TB = (Tidak Baik). Dengan bobot penilaian untuk setiap jawaban tersebut adalah SB = 5 ; B = 4 ; CB = 3 ; KB = 2 ; TB = 1

#### 2. Tabulasi Data

Angket yang dibuat kemudian dibagikan kepada responden. Teknik pemilihan responden (sampel) dilakukan dengan metode *purposive sampling* yaitu pemilihan sampel dengan cara menentukan responden berdasarkan kriteria tertentu yang disesuaikan, pada penelitian ini responden yang digunakan adalah petugas Balai Konservasi Sumber Daya Alam Provinsi Bengkulu dengan 10 sampel. Sebelum melakukan perhitungan dengan menggunakan skala *likert*, terlebih dahulu dilakukan pencarian intervalnya dengan persamaan

$i = \frac{m-n}{k}$	<p>Keterangan :</p> <p>i = interval kelas</p> <p>m = angka tertinggi skor</p> <p>n = angka terendah skor</p> <p>k = banyak kelas</p>
---------------------	--

Gambar 5.42 Persamaan skala *likert*

Dengan  $i = 0,8$  yang didapatkan dari  $m=5$ ,  $n=1$  dan  $k=5$  serta dipilih ketetapan skala terendah adalah 1,00, maka kategori penilaian yang dihasilkan adalah sebagai berikut :

Tabel 5.18 Kategori Penilaian

Interval	Kategori
4,24 - 5,00	Sangat baik
3,43 - 4,23	Baik
2,62 - 3,42	Cukup Baik
1,81 - 2,61	Kurang baik
1,00 - 1,80	Tidak baik

Kemudian data dengan skala *likert* dianalisis dengan menghitung skor pada tiap-tiap interval dari pernyataan yang diberikan ke responden. Berikut ini adalah hasil penilaian dari pengujian terhadap pengguna untuk masing - masing variabel tampilan, kemudahan pengguna, dan kinerja sistem:



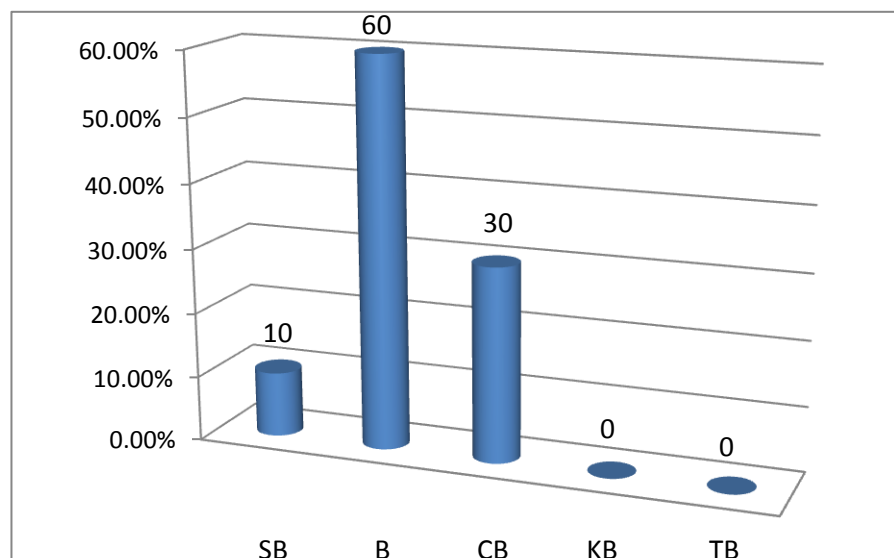
1) Variabel Tampilan

Untuk penilaian variabel tampilan didapatkan hasil seperti pada tabel dibawah ini.

Tabel 5.19 Hasil Penilaian Variabel Tampilan

No.	Tampilan (V1)	M	Frekuensi Jawaban				
			SB	B	CB	KB	TB
1	Komposisi warna	3.70	1	5	4	0	0
2	Kejelasan teks yang ada	3.80	1	6	3	0	0
3	Tata Letak Tampilan	4.00	2	6	2	0	0
4	Variasi tampilan	3.60	0	6	4	0	0
5	Kualitas Tampilan	3.90	1	7	2	0	0
	Jumlah frekuensi jawaban		5	30	15	0	0
	Presentase rata-rata		10.00%	60.00%	30.00%	0%	0%
	Total rata-rata kategori	3.80					
	Kategori	<b>"BAIK"</b>					

Dari tabel tersebut terlihat bahwa penilaian terhadap variabel 1 memiliki nilai rata-rata 3.80. Berdasarkan kategori penilaian pada tabel 5.17 nilai rata-rata 3.80 berada dalam interval 3.43 – 4.23. jadi dapat disimpulkan bahwa penilaian pada variabel 1 termasuk kategori “Baik”.



Gambar 5.43 Grafik Presentase Hasil Angket Variabel 1

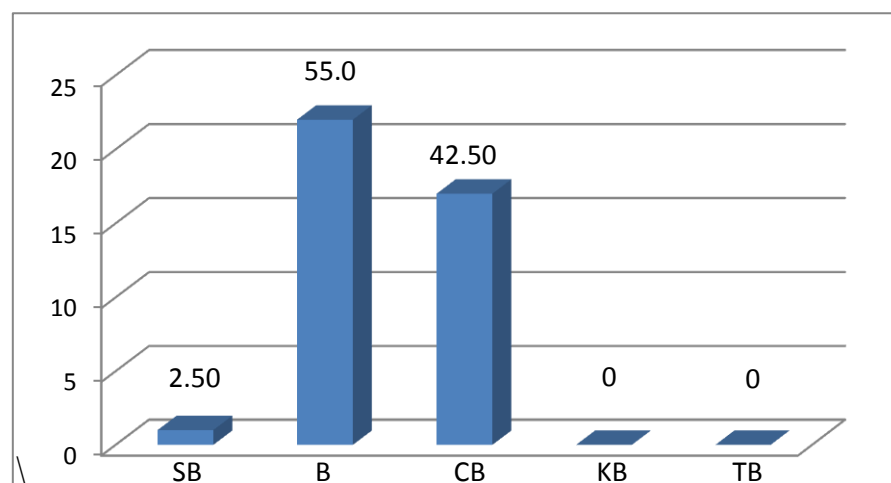
## 2) Variabel Kemudahan Pengguna

Untuk penilaian variabel kemudahan pengguna didapatkan hasil seperti pada tabel dibawah ini.

Tabel 5.20 Hasil Penilaian Variabel Kemudahan Pengguna

No.	Kemudahan Pengguna (V2)	M	Frekuensi Jawaban				
			SB	B	CB	KB	TB
1	Kemudahan intruksi yang diterapkan	3.50	0	5	5	0	0
2	Kemudahan mengoperasikan sistem	3.50	0	5	5	0	0
3	Kemudahan memahami informasi yang diberikan	3.60	0	6	4	0	0
4	Kemudahan menginstall sistem	3.80	1	6	3	0	0
	Jumlah frekuensi jawaban		1	22	17	0	0
	Presentase rata-rata		2.50%	55.00%	42.50%	0%	0%
	Total rata-rata kategori	3.60					
	Kategori	<b>"BAIK"</b>					

Dari tabel tersebut terlihat bahwa penilaian terhadap variabel 2 nilai rata-rata 3.60. Berdasarkan kategori penilaian pada tabel 5.17 nilai rata-rata 3.60 berada dalam interval 3,43 - 4,23. jadi dapat disimpulkan bahwa penilaian pada variabel 2 termasuk kategori “Baik”.



Gambar 5.44 Grafik Presentase Hasil Angket Variabel 2

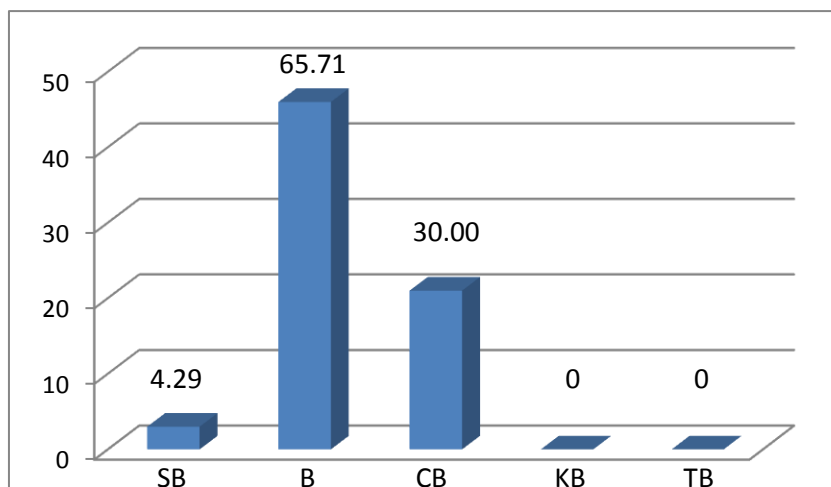
### 3) Variabel Kinerja Sistem

Untuk penilaian variabel kinerja sistem didapatkan hasil seperti pada tabel dibawah ini.

Tabel 5.21 Hasil Penilaian Variabel Kinerja Sistem

No.	Kinerja Sistem (V3)	M	Frekuensi Jawaban				
			SB	B	CB	KB	TB
1	Tujuan/Kegunaan Sistem	3.80	1	6	3	0	0
2	Urutan penyajian pada sistem	3.90	1	7	2	0	0
3	Fasilitas/Fitur-fitur dalam sistem	3.10	1	6	3	0	0
4	Kecepatan waktu akses sistem	3.70	0	7	3	0	0
5	Pengolahan data yang tersedia	3.60	0	6	4	0	0
6	Ketepatan informasi sistem	3.60	0	6	4	0	0
7	Kesesuaian hasil informasi dengan kebutuhan pengguna	3.80	0	8	2	0	0
	Jumlah frekuensi jawaban		3	46	21	0	0
	Presentase rata-rata		4.29%	65.71%	30.00%	0%	0.00%
	Total rata-rata kategori	3.70					
	Kategori	<b>"BAIK"</b>					

Dari tabel tersebut terlihat bahwa penilaian terhadap variabel 3 memiliki nilai rata-rata 3.70. Berdasarkan kategori penilaian pada tabel 5.17 nilai rata-rata 3.70 berada dalam interval 3.43 - 4,23. jadi dapat disimpulkan bahwa penilaian pada variabel 3 termasuk kategori “Baik”.



Gambar 5.45 Grafik Presentase Hasil Angket Variabel 3

## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Berdasarkan proses yang telah dikerjakan dimulai dari Definisi dan Analisis Kebutuhan, Desain Sistem dan Perangkat Lunak, Integrasi dan Pengujian Sistem, serta Operasi dan Pemeliharaan. Adapun kesimpulan yang didapat dari penelitian ini adalah :

1. Sistem Informasi Georafis Sebaran Flora, Fauna dan Aktivitas Konservasi dalam Mendukung Kelestarian Alam Berbasis Manajemen *Resort* Provinsi Bengkulu memanfaatkan teknologi *GPS* dan *Google Maps API*, yang terintegrasi antara perangkat *Mobile Android* dan *Web* sehingga informasi yang didapatkan lebih cepat dan tepat. Hasil keluaran sistem berupa lokasi dan peta aktivitas konservasi. Sistem juga mampu menampilkan rute serta statistik untuk memudahkan pihak yang terkait dalam memperoleh gambaran lokasi dan aktivitas konservasi.
2. Berdasarkan hasil pengujian kelayakan, implementasi, dukungan dan relevansi sistem ini terhadap kelestarian alam dan lingkungan hidup di provinsi Bengkulu merujuk pada hasil validasi dari tiga variabel angket (tampilan, kemudahan pengguna dan kinerja sistem), sistem dikategorikan **baik** atau **layak** dengan presentase 59.24 % hasil rerata tiap variabel.

## 6.2 Saran

Berdasarkan pembahasan yang telah diuraikan, saran dari penelitian ini adalah sebagai berikut :

### 1. Bagi pengguna

Sistem yang dibangun bekerja dengan baik pada *mobile phone* berbasis *android* dan *web*. Performansi sistem sangat tergantung pada kecepatan akses internet yang ada. Sehingga dibutuhkan akses internet yang cukup cepat dan stabil serta *provider* yang sesuai pada daerah penggunaan.

### 2. Bagi pengembang sistem

- 1) Pengembangan sistem lebih lanjut diharapkan dapat diimplementasikan pada *Operating System* lainnya.
- 2) Sistem ini hanya merujuk pada informasi aktivitas konservasi yang berjalan pada Balai Konservasi Sumber Daya Alam (BKSDA) Provinsi Bengkulu dan belum dilengkapi dengan fasilitas yang memungkinkan untuk penerapan sistem untuk provinsi lainnya. Sehingga ada kemungkinan perubahan berdasarkan kebutuhan apabila hendak diterapkan untuk provinsi lainnya maupun skala sistem yang lebih besar. Untuk tahap pengembangan selanjutnya diharapkan agar dapat dilengkapi dengan fasilitas pengembangan sistem yang lengkap.

## DAFTAR PUSTAKA

- Alesheikh, Helali. 2002. *Web GIS : Technologies and Its Application*. Proceeding of Digital Earth 2002, Canada.
- Al-Bahra bin Ladjamudin. 2005. *Analisis dan Desain Sistem Informasi*. Yogyakarta: Graha Ilmu.
- Andri Kristanto. 2007. *Perancangan Sistem Informasi Dan Aplikasinya*. Penerbit Gava Media. Klaten
- Anonim. 2013. *Discover Bengkulu The Land of Rafflesia*. Bengkulu : Dinas Kebudayaan dan Pariwisata Provinsi Bengkulu.
- Anonim. 2010. *Pengelolaan Taman Nasional Berbasis Resort*. Makassar: Balai Taman Nasional Alas Purwo.
- Aronoff S. 1989. *Geographic Information Systems: A Management Perspective*. WDL Publications. Ottawa.
- Bennet, Simon, Mcrobb, Steve, Farmer, Ray. 2002. *Object Oriented Analysis and Design Using UML*. McGraw-Hill Book Co., New York.
- Bodnar, George H, and William S.Hopwood. 2006. *Sistem Informasi Akuntansi*, Buku I. Jakarta: Penerbit salemba empat.
- Budi, Heri dan Wahyu. 2008. *Sistem Informasi Geografis Penyebaran Penduduk Berdasarkan Tingkat Usia di Kabupaten Selman Berbasis Web*. Jurusan Teknik Informatika, UPN Veteran Yogyakarta: Jurnal Seminar Nasional Informatika 2008.
- Dharwiyanti, Sri. 2003. *Pengantar Uniified Modelling Language (UML)*. [www.ilmukomputer.com](http://www.ilmukomputer.com).
- Edi, Sena dan Tri. 2011. *Pengembangan Sistem Informasi Geografis (SIG) Pada Platform Google Untuk Penanggulangan Kebakaran di Jakarta Selatan*. Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Bina

Nusantara Jakarta : Jurnal Seminar Nasional Aplikasi Teknologi Informasi 2011.

Erin, Arna dan Rengga. 2011. *Aplikasi Sistem Informasi Geografis Tentang Kos-Kosan Menggunakan Google Maps API 3 (Studi Kasus : Surabaya bagian timur)*. Surabaya: Jurnal Politeknik Elektronika Negeri Surabaya, Institut Teknologi Sepuluh Nopember.

Handoko, Sutra. 2010. *Manajemen Basis Data Spasial Kawasan Hutan Beserta Ekosistemnya Wilayah Provinsi Bengkulu Dengan Menggunakan SIG Berbasis Web*. Bengkulu: Skripsi Prodi Teknik Informatika Universitas Bengkulu.

Helali. 2001. *Design and Implementation of a Web GIS for the City of Tehran*. Department of Geodesy and Geomatics Engineering K.N Toosi University of Technology Tehran. Iran

Husein Umar. 2003. *Metodologi Penelitian Untuk Skripsi dan Tesis Bisnis*. Jakarta: PT. Gramedia Pustaka.

IUCN. 2003. *Guidelines for Management Planning of Protected Areas*. UK: International Union for Conservation of Nature and Natural Resources, World Commision on Protected Areas. UK: Cadriff University.

Jogiyanto. 2005. *Analisis dan Desain Sistem Informasi*. Yogyakarta: Andi.

Keraf, Gorys. 1997. *Komposisi: Sebuah Pengantar Kemahiran Bahasa*. Ende-Flores: Nusa Indah. Pusat Pembinaan dan Pengembangan

Lucky. 2008. *XML Web Service: Aplikasi Desktop, Internet, Handphone*. Indonesia: Jasakom

MacKinnon J, Mackinnon K, Child G, Thorsell J. 1993. *Pengelolaan Kawasan yang Dilindungi di Daerah Tropika*. Yogyakarta : Gadjah Mada University Press.

- Mulyatiningsih, Endang. 2012. *Metode Penelitian Terapan Bidang Pendidikan*. Bandung: Alfabeta.
- Nugroho, Adi. 2005. *Analisis dan Perancangan Sistem Informasi Dengan Metodologi Berorientasi Objek*. Informatika, Bandung.
- Prahasta, Eddy. 2009. *Sistem Informasi Geografis*. Bandung: Informatika.
- Pratama, Haris. 2012. *Rancang Bangun Aplikasi Sensus Penduduk Di Kota Bengkulu Berbasis Mobile Android Dengan Web Service*. Bengkulu: Skripsi Prodi Teknik Informatika Universitas Bengkulu.
- Pressman, R.S. 2002. *Rekayasa Perangkat Lunak, Indonesian Translation Copyright, Penerbit ANDI and McGraw-Hill Book, Co., Yogyakarta*
- Safaat H, Nazruddin. 2011. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC berbasis Android*. Bandung: Informatika.
- Setyadi, P.G.I, Suadi, W., dan Mazharuddin, A. 2011. *Implementasi Web Service Di System Android Menggunakan Python Dan SL4A (Scripting Language For Android)*. ITS, Surabaya.
- Sommerville. 2003. *Software Engineering (Rekayasa Perangkat Lunak)*. Jakarta: Erlangga.
- Sugiyono. 2005. *Memahami Penelitian Kualitatif*. Bandung: Alfabet.
- Sukmadinata. 2006. *Metode Penelitian Pendidikan*. Bandung: Remaja Rosdakarya.
- Tulach, Jaroslav. 2008. *Practical API Design: Confessions of a Java™ Framework Architect*. Apress, United States of America.